Australian Government
Department of Defence
Defence Science and
Technology Organisation

Detection and Tracking of Corner Points for Structure from Motion

Tristrom Cooke and Robert Whatmough

Intelligence, Surveillance and Reconnaissance Division
Defence Science and Technology Organisation

DSTO–TR–1759

ABSTRACT

This report describes the first stage in solving the structure from motion problem, which is to detect feature points and track them from frame to frame. A number of detectors from the literature, as well as some specially developed detectors, are assessed using a fly-over sequence of Parafield control tower. It is found by this measure that the Harris detector is the best of the conventional detectors, and that two new detectors (the Generalised Hough transform and covariance tracking based methods) appear to give even better results for many cases. Finally, a method for detecting corners by fusing the outputs of numerous detectors is described.

Published by

Defence Science and Technology Organisation
PO Box 1500
Edinburgh, South Australia 5111, Australia

Telephone: (08) 8259 5555
Facsimile: (08) 8259 6567

AR No. 013-476
1st August 2005

APPROVED FOR PUBLIC RELEASE

Detection and Tracking of Corner Points for Structure from Motion

EXECUTIVE SUMMARY

The motivating problem behind this report is to extract a 3D structure from motion based on a video sequence of a building from an airborne sensor. This has application to site monitoring and surveillance, targeting and operations planning. Structure from motion problems have been studied for some time, and the established methods of solution usually consist of four major steps. The first step is to extract useful features (points, lines, or planes) from each of the images. Then, each of the features is tracked throughout the image sequence. Following this, a 3D model and camera motion model are determined which are consistent with the measured feature positions. Finally, high level information about the scene (such as the general structure of buildings usually consisting of plane facets intersecting perpendicularly) is used to refine the model.

The current report describes aspects of the first two steps in structure from motion; specifically, the detection and tracking of point features.

As an aid to the reconstruction of a wire-frame model of a building, corner points corresponding to the intersection of three or more building facets are the most useful. The first section of the report extensively analyzes a large number of standard and new corner detection techniques.

Once the features have been detected, they must be tracked between image frames. The second half of the report provides a quick overview of tracking methods, with a focus on those aspects of the problem which are unique to tracking features in video sequences. One focus of this part of the report is on the relationship between trackers and feature detectors, and this is used to create new corner detectors. Some possible methods for fusing the outputs of corner detectors to further improve performance have also been examined.

Throughout the report, stand-alone detectors, tracking based detectors and fused detectors have been described. All of these detectors have been evaluated using a thermal IR video sequence of the Parafield airport control tower. The performance of the commonly used detectors from the literature was found to be consistently highest for the Harris corner detector. Two new corner detectors described in this report, the Generalised Hough Transform and the covariance tracking based detectors, were found to give better results than the Harris detector in a number of cases. It is expected that these results will also extend to other types of video imagery.

Authors

Tristrom Cooke
ISRD

This author obtained a B.Eng (Hons) in Electrical Engineering from the University of South Australia in 1992, a B.Sc (Hons) in Applied Mathematics from the University of Adelaide in 1995, and completed a PhD in Applied Mathematics (also at the University of Adelaide) in the area of thermo-elasticity at the end of 1998. He was then employed by CSSIP (CRC for Sensor Signals and Information Processing) until 2005, where he has mostly worked on projects relating to recognition of targets in both SAR and ISAR radar imagery. He is now full time employee in ISRD of DSTO, where he is working on structure from motion.

Robert Whatmough
ISRD

Robert Whatmough received his B Sc (Hons) at Monash University in 1969. After many years in the old Computing Services Group he moved into the image processing field, first in Optoelectronics Division and then in several others as reorganisations came and went.

Since then he has worked on restoration, enhancement, classification and registration of images, aerial image prediction, object matching and shape inference. His present interests include enhancement of and shape inference from video sequences.

Contents

Appendix

Figures

1 Introduction

The aim of structure from motion is to recover a 3D model of a scene from a video sequence taken using a moving camera. In this report, the problem is examined from the perspective of extracting models of buildings from an airborne video sensor as in a UAV. The particular example for which data is available is a low resolution fly-over of the Parafield airport control tower. Although the methods described and developed in this report are applied only to this sequence, it is hoped that they perform equally well with other similar types of imagery.

Structure from motion problems have been studied for some time, and the established methods of solution usually consist of four major steps. The first step is to extract useful features (points, lines, or planes) from each of the images. Then, each of the features is tracked throughout the image sequence. Following this, a 3D model and camera motion model are determined which are consistent with the measured feature positions. Finally, high level information about the scene (such as the general structure of buildings usually consisting of plane facets intersecting perpendicularly) is used to refine the model.

The current report describes aspects of the first two steps in structure from motion; specifically, the detection and tracking of point features. The most frequently detected feature for most structure from motion problems are corners. Certainly, in order to reconstruct a wire-frame model of a building, the corners corresponding to the intersection of three or more building facets are crucial. Section 2 provides an extensive analysis of a large number of standard and new corner detection techniques, and evaluates their capabilities in the context of structure from motion. Lines and line segments are also useful, but have been less extensively studied. An overview of some of the line detection techniques will be provided in a follow-up report.

Once the features have been detected, they must be tracked between image frames. The tracking problem has been examined in excruciating detail in the literature, but most of the work is outside the scope of this report. Therefore, Section 3 gives only a quick overview of some of the tracking methods, with a focus on those aspects of the problem which are unique to tracking features in video sequences. This section tries to focus on the relationship between trackers and feature detectors, and uses this to create new corner detectors. Section 4 then describes methods for fusing the outputs of corner detectors to improve performance, Some conclusions and recommendations for future work are then summarised in Section 5.

2 Point detection

There is a great deal of literature on point detection algorithms, but in practice many vision applications rely on the Harris detector. This uses a heuristic measurement which increases where the image has large image gradients in two perpendicular directions. The maximum of the Harris detector is usually not exactly on the corner, and this lack of consistent localisation has lead to other detectors being used in some applications. A more complete discussion of the Harris detector is given in Subsection 2.1, and Subsection 2.2 describes some similar detectors which assume that the image can be modelled by a continuous function, such as the Shi-Tomasi detector which is currently implemented in ADSS.

Another commonly cited detector, based on a completely different approach is called SUSAN. This detector effectively performs a local fuzzy segmentation of the points based on their intensity, and uses the size of the segments as a measure of cornerness. A more complete description of the SUSAN corner detector is provided in Section 2.3.

Recently, another detector called SIFT has received a lot of attention. This is specifically an interest point detector rather than a corner or junction detector. SIFT searches for blobs in imagery using a multiscale framework. Each detected feature is also associated with an invariant shape key, which may be used to find correspondences between features detected in different images. A brief description of SIFT is described in 2.4.

A number of other less commonly used and some new detector algorithms are also described. Subsection 2.5 describes the local energy detector, which was inspired by the structure of biological vision systems. Some parallels between the local energy and the function based prescreeners are made, and some modifications described for improving the corner detection performance. Subsection 2.6 provides details of a new type of corner detector based on the accumulation of evidence that a point is at the intersection of two edges. This accumulation method is related to the computation of the Hough transform.

Having defined a large number of corner detectors, some methods are needed to evaluate them in the context of structure from motion. Subsection 2.7 summarises some of the existing performance measures, and then evaluates each of the corner detectors for their performance on the Parafield fly-over data. These performance measures are also used later in Section 3 to assess corner detectors based on trackers, and in Section 4 which considers fusing corner detectors to improve their performance.

2.1 The Harris detector

The Harris detector (sometimes known as the Stephen-Harris or the Plessey detector) is probably the most widely used point detector in the vision literature. It is based on an idea by Moravec [18], where points of interest corresponded to large intensity gradients in all directions. To implement this, Moravec computed a local correlation between the original image and copies of the image shifted in each of the four image directions. The interest point was detected if the largest difference in correlation was above some threshold. Harris and Stephens [6] extended this detector by reformulating the problem using image derivatives. Also, a Gaussian smoothing operator was used to localise the image feature instead of a square window. This had the advantage of making the detector insensitive to rotation. The final detector was based on an "inspired formulation" [6], and is defined as follows.

Let I_x, I_y be the derivatives of the image I in the x and y directions respectively. We define three quantities g_x, g_y and g_{xy} to be the images I_x^2, I_y^2 and $I_x I_y$ convolved with a zero mean Gaussian with width σ (which is often set to 1). Then the measure of cornerness is given by

$$g_x g_y - g_{xy}^2 - \alpha(g_x + g_y)^2 \tag{1}$$

for some constant α. The value of α was not specified in the original paper, but later papers have used α about 0.04 (although Schmid et al. [24] used $\sigma = 2$ and $\alpha = 0.06$). The optimal value for α has been investigated by Rockett [22], who examined the ability of the detector to distinguish images centered on corners from ones where the corners were shifted by a pixel from the centre (given the name ‘ non-obvious non-corners ’). Rockett used the Harris detector to classify the data, and measured the area under the ROC curve to determine the performance for a range of α. For $\sigma = 1$, the performance was found to be very constant over the range $\alpha \in (0.04, 0.06)$ with the maximum occurring somewhere in the middle.

Several papers have been published concerning evaluation of the performance of corner and interest point detectors. Rockett's work [22] was based on simulated data sets, containing $10,000$ instances of corners and non-corners. Most of the non-corners were modelled by either edges or areas of roughly constant intensity. All of the examples had been corrupted by noise, and subjected to diffraction effects and other processes designed to mimic the entire optical imaging process. Detectors were then applied to the resulting images, and the area under the ROC curve as the performance measure. It was found that the Harris detector produced significantly better discrimination between corners and edges and/or image patches than two other detectors (the Kitchen-Rosenfeld detector [11], which is based on second order image derivatives, and the Paler detector [20], which is based on order statistics). However, Rockett also measured the localisation accuracy which was the area under the curve obtained for distinguishing the response centered on corners, and where the corners were shifted by one pixel. By this measure, the Harris detector was the worst performing detector.

An earlier paper, Schmid et al. [24], used repeatability of the detection of features as a performance measure. This was measured by detecting points in one image frame, and then applying an affine transformation (rotation as well as scaling), adding noise, or varying the illumination to produce a new image. The corner detection was applied again, and the percentage of consistent detections measured. In this way, the paper compared five detectors (those by Harris, Cottier, Horaud, Heitger and Förstner). Of these, only the Harris is frequently used in computational vision, although the Heitger detector [7] (which is based on Gabor filters for finding edges in different directions, and combining the edge images into a corner image) seems more frequently referred to for biological applications. It was found that the Harris detector consistency gave the most repeatable set of detections.

With the aim of improving the repeatability of detections in two images differing by an affine transformation, Mikolajczyk and Schmid [17] suggested some modifications to the standard Harris. Scale invariance is achieved by combining two separate interest point detectors. The first is based on the Laplacian of the Gaussian (LoG), first suggested by Lindeberg [12] for detecting blobs (this was later refined by Lowe to become the SIFT detector, described in Subsection 2.4). This examines the response of the image to convolution with LoGs of various widths. The scale of the image at that point corresponds to the width at which the response is a maximum. Once the scale is determined, the Harris detector (which provides the corner measure) is applied to the image, with the value of

σ specified by the image scale at each point. From the scale independent detector, affine invariance was produced by estimating the affine transformation which maximises the local isotropy (given by the ratio of smallest to largest eigenvalues for the Karhunen Loeve transform of neighbouring pixels). Although this appears to give more consistent matches, the points detected appear more like blobs than corners, and so might be of less utility for 3D reconstruction than points from the original Harris detector.

2.1.1 The GA trained Harris detector

The performance of the Harris detector is dependent on α, and on the shape of the distribution with which the images are convolved. The choice of a Gaussian seems somewhat arbitrary, so it was attempted to find an alternative which would be optimal with respect to the performance on a simulated training set. Since this performance will be subject to statistical fluctuation, many standard optimisation techniques could not be used. Instead, a genetic algorithm (GA), as described in the Appendix, was constructed and applied to

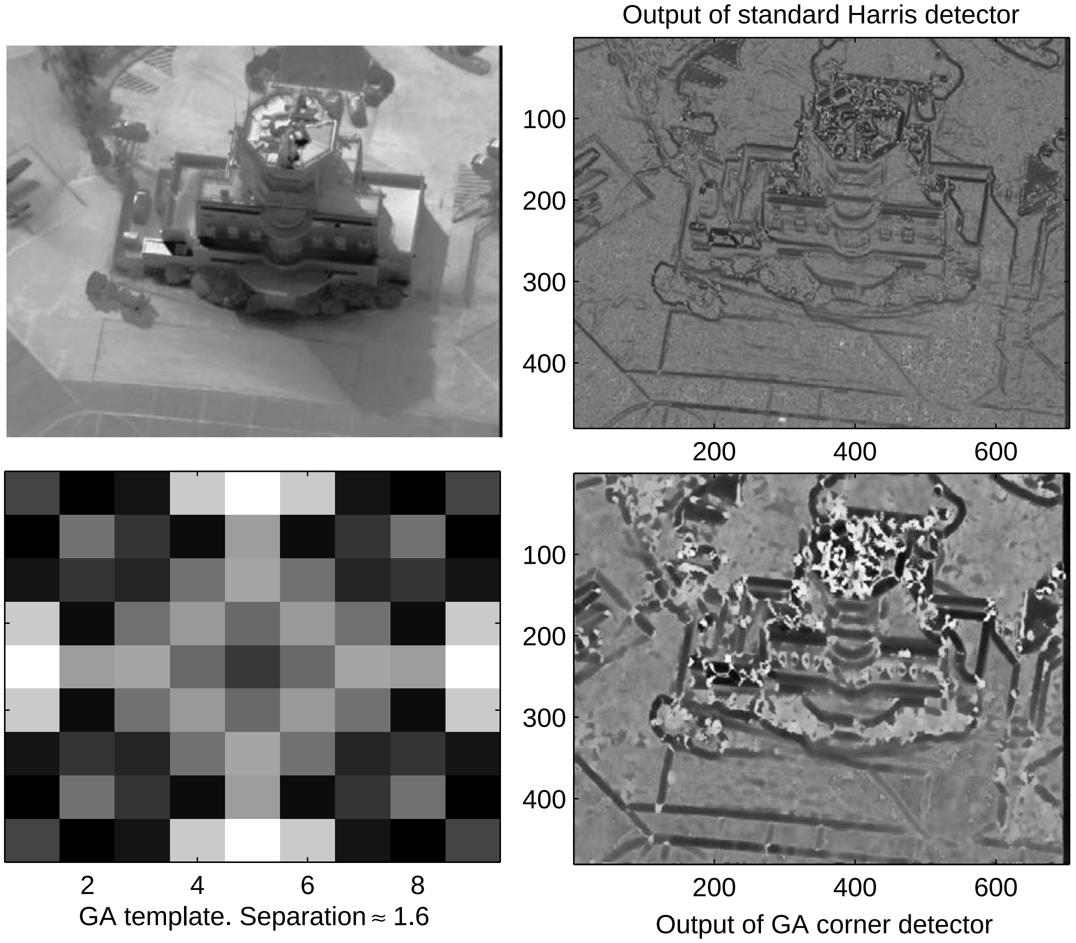

Output of standard Harris detector
100
200
300
400
200
400
600
2
4
6
8
2
4
6
8
GA template. Separation ≈ 1.6
100
200
300
400
200
400
600
Output of GA corner detector

Figure 1: *Output from the Harris detector, with and without the modified convolution function*

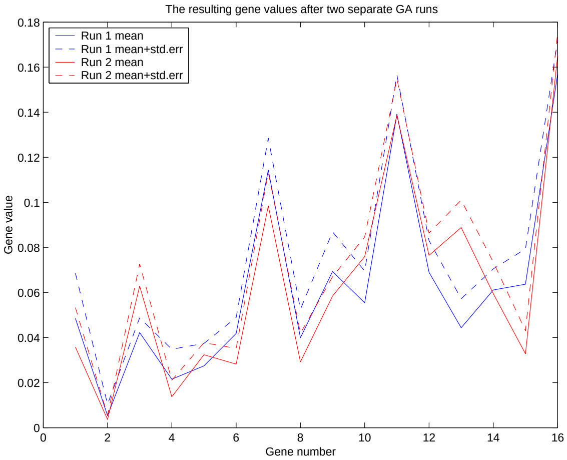


Figure 2: *Output from two different runs of the GA*

find both of these parameters. The convolution function G, was defined over an 11×11 grid, and forced to be invariant to flipping the coordinate axes so that the resulting detector would be as well. This made a total of 15 unknown function parameters, and the value of α to be optimised with respect to the reward function.

Initially, the reward function for the GA was based on a data set consisting of corners (of various angles) and non-corners (mostly edges). The data sets were corrupted by additive white noise (with size chosen to be one fifth of the edge strength), and the detector performance was measured by the Fisher separation between the closest corner and non-corner classes. Figure 1 shows an image taken from the Parafield fly-over, the output from the original Harris detector, and the modification of the Harris to maximise the Fisher separation.

The output template from the genetic algorithm was reasonably consistent between runs, as can be seen from the two example genotypes in Figure 2. These two examples show the mean and the standard deviation in the population after the GA no longer seems to be improving performance. While a few of the variables showed noticeable differences between runs, the general shape of the function appears consistent. Performance-wise, the results were almost identical, and for each of the 10 times that the GA was reinitialised and run, the Fisher separation between the corner and the edge classes increased from about 0.7 for the original Harris detector to about 1.6.

To give a more useful measure of performance of the new detector, a larger data set consisting of 10,000 instances of each of the corner and edge types from the left of Figure 3 were computed. The original and GA versions of the corner detector were then applied to each of the sets, and ROC curves were computed, as shown in the right of Figure 3. The area under the curve was 0.995 for the GA Harris, but only 0.911 for the original. Also, for a 90 percent corner detection rate, the GA modification reduced the percentage of false alarms by a factor of 50. It should be mentioned however that there may have been some over-training specifically for the corner types in the data set (as evidenced by the strong diagonal values of the template shown in Figure 1. The performance gain may not be as great when a more continuous range of corner angles are used in training. Secondly,

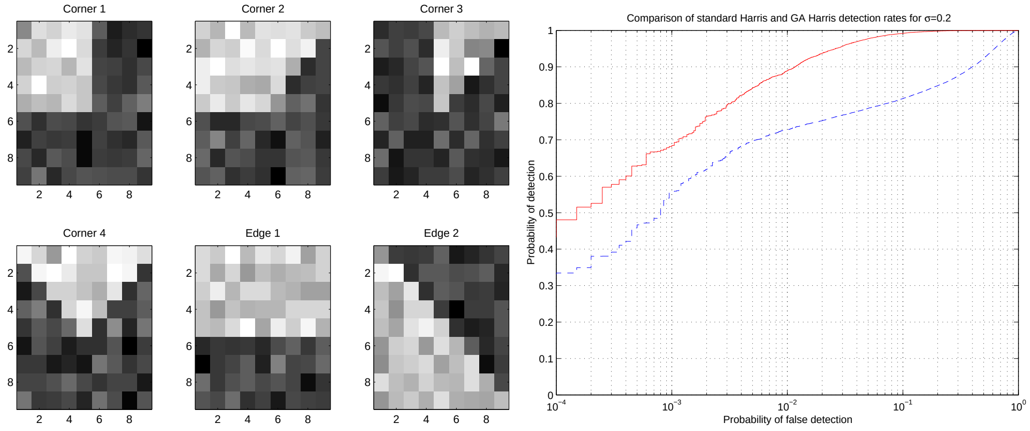


Figure 3: *Performance of the original and the GA Harris detectors on a simulated data set*

optical imagery has perhaps less problem with additive white noise than with blurring and diffraction effects. Therefore, although the new detector exhibits insensitivity to speckle, it may not prove a better detector in real images. Thirdly, as can be seen by the output of Figure 1, the peaks in the detection map of the image are broad, so the localisation error will be significantly increased.

The area under the ROC curve on a synthetic data set is not the only measure of success that can be used to train the genetic algorithm. Some other reward measures have also been tested, and these are summarised in Section 2.7.

2.2 The Shi-Tomasi and related detectors

The Harris detector is only one of an entire class of detectors which approximate the image by a continuous function, and uses estimates of the derivatives to detect corners. The most commonly used of these is the Shi-Tomasi detector [25], which is the basis of the commonly used KLT (Kanade-Lucas-Tomasi) tracker, as described in Section 3.2. The KLT tracker requires the solution to the matrix equation

$$\left(\int \nabla I(\mathbf{x}) \nabla^T I(\mathbf{x}) w(\mathbf{x}) dx \right) \mathbf{d} = \mathbf{e}$$

where $I(\mathbf{x})$ is the image, ∇ is the gradient operator, defined as a column vector, $w(x)$ is some local weighting function (frequently a constant over some rectangular window), $\mathbf{d}$ is the displacement between the images, and $\mathbf{e}$ is a measure of the dissimilarity of the images. The image displacement can be found most accurately when the matrix on the left has large eigenvalues, which do not differ in scale too much. As a result, if λ_1 and λ_2 are the two eigenvalues, the best corners for tracking have been chosen to satisfy

$$\min(\lambda_1, \lambda_2) > t$$

for some threshold t. Therefore, the term on the left hand side can be taken as a local measure of cornerness. As with the Harris detector, peaks will occur where there is a large intensity variation in perpendicular directions.

The detector proposed by Harris and Stephens was as given in equation (1). A paper by Noble [19] however cites another earlier paper by Harris in which he incorrectly claims that the corner detector was formulated as

$$\frac{g_x g_y - g_{xy}^2}{g_x + g_y} \tag{2}$$

When the threshold of the Harris corner measure is zero, this will be equivalent to thresholding (2) at the value α, but thresholds at other levels are not equivalent. It is not known whether this reformulation was instigated by Noble for the purpose of removing the empirically chosen constant α (as is sometimes claimed). In any case, Noble analysed the detector, and showed that it was related to the curvature of the manifold defined by $(x, y, I(x, y))$ in the plane perpendicular to the gradient. For this reason, it will be referred to here as the Noble detector. The following modification of the Noble detector

$$\frac{g_x g_y - g_{xy}^2}{(g_x + g_y + \sigma^2)^2} \tag{3}$$

can be made to this detector to make it largely invariant to the intensity scale of the image. The parameter σ^2 is related to the level of noise in the imagery, so that statistical fluctuations in pixel intensities will rarely produce a strong corner response. This formula will be referred to as the intensity invariant Noble detector.

Another detector which has seen frequent use (owing mostly to being one of the earliest corner detectors) is the Kitchen and Rosenfeld detector [11]. It basically models the intensity as a continuous function, and estimates the product of curvature of the intensity contour line at that point in the image, and the edge strength. This gives the rotation invariant measure of cornerness,

$$\frac{I_{xx} I_y^2 + I_{yy} I_x^2 - 2 I_{xy} I_x I_y}{I_x^2 + I_y^2}, \tag{4}$$

where I_x and I_{xx} are the first and second derivatives of the image with respect to the x variable. If there is noise in the imagery, the estimates for the second derivatives can be quite poor which may lead to spurious corner detection.

Tissainayagam and Suter [28] assessed the performance of four types of corner detectors (Noble [1], Kitchen-Rosenfeld, SUSAN, and the KLT (Kanade-Lucas-Tomasi) method) for feature tracking applications. Here, the tracking capability of the detector is measured by

[1] This is referred to by the paper as the Harris detector, but specified that the corner measure is of the form of equation (2) rather than equation (1)

considering video sequences of a static scene. An original set of detections are produced for the first frame, and these are tracked throughout the entire image sequence. The performance of the detector is measured by the percentage of points that can be tracked throughout the sequence, and by the mean distance of the tracked point from the original detection. By these measures, it was found that the Noble detector and the KLT method were significantly better than the other three detectors in all four of the test sequences. The Noble detector proved better than the KLT method in half of the sequences. It is not really useful to compare these two methods however, since they both use not only different corner detectors, but different methods of tracking, and it cannot be certain whether that it is the detector or the tracker which is producing the difference in performance.

2.3 The SUSAN detector

The SUSAN detector, invented by Brady and Smith [27], is one of the most frequently cited corner detectors in more recent literature. It stands for Smallest Univalue Segment Assimilating Nucleus, and is based on extracting a segment of a small circular image mask which has roughly the same intensity as the central pixel (or nucleus). This segment (called the USAN) will have an area which is smallest at a corner of two regions of unequal intensity (about one quarter of the total area for a perpendicular corner), increase to one half for an edge, and become the entire area in the centre of a region with uniform intensity. The inverse of this area at each point can therefore be used as a measure of cornerness.

Due to image noise and intensity variations, there is no obviously correct method for determining whether two pixel's intensities are equivalent. Brady and Smith's approach is to require a user defined threshold t and defines a fuzzy measure of compatibility given by

$$c(x, y) = \exp\left[-\left(\frac{I(x) - I(y)}{t}\right)^J\right]$$

for some even integer J, and threshold t. When $J = 2$ this is a Gaussian, and as $J \to \infty$ the fuzzy measure becomes a hard $0 - 1$ decision. Each pixel in the image mask then contributes some amount to the area of the USAN. For a given value of t, the performance of SUSAN for edges was then measured by looking at the response due to two classes of image (an edge of strength dt and a uniform patch) with the addition of white noise of standard deviation σt. If the responses for the classes have means μ_e, μ_p and variances σ_e^2, σ_p^2, then the degree of overlap of the two classes, given by

$$F(J, d, \sigma) = \frac{\sigma_e + \sigma_p}{\mu_e - \mu_p},$$

should be minimised to give a better performance. Instead of choosing the maximum for particular values of d and σ, a uniformly weighted average performance over a range of likely values was calculated to give

$$f(J) = \int_{d=1}^{10} \int_{\sigma=0}^{1/\sqrt{2}} F(J, d, \sigma) d\sigma dd.$$

This performance measure was then calculated over a range of J, and it was found that a value of $J = 6$ gave the least amount of overlap (with a value for F of 0.850 compared to about 0.87 as $J \to \infty$, which is not a huge improvement).

Once a value of cornerness has been calculated for each point in the image, some post-processing is applied to remove false alarms. For each detected corner (*i.e.* one for which the area of the USAN is less than one half), it is checked whether there exists a radial line from the nucleus for which every point belongs to the USAN. This type of processing effectively removes small blobs from being detected as corners.

In the Parafield fly-over imagery, there are many cases where connected facets have a visible edge, but there is not a large intensity difference between them. SUSAN has not been designed to work in these cases, and so misses many of these corners. SUSAN also seems to work poorly in other situations too. For instance, in Suter and Tissainayagam's paper [28], SUSAN was used to track the hundred strongest detected points throughout a video of some static scenes. The number of corners that were successfully tracked were then plotted as a function of the number of frames. It was found that of the corner detectors tested, SUSAN consistently produced the lowest number of stable corners, and also gave the worst localisation of those corners which were successfully tracked. SUSAN however is quite different from the other detectors tested in that there is not really any idea of a dominant corner. In any given frame, any corner could produce a stronger response than any other corner, even though the corners might produce consistently stronger responses than non-corners. In that scenario, if the actual number of corners was greater than 100, any given corner might not be consistently tracked by SUSAN despite effectively removing any non-corners.

Another paper by Martinez-Fonte *et al.* [15] produces empirical performance measures of corner detection performance on IKONOS satellite imagery of buildings from the city of Ghent. This was done by manually selecting corner points from an image, and running the corner detector over the same image. For each threshold on the corner detector, a probability of detection and a false alarm rate are obtained, and a ROC curve can be plotted. In this paper, the Noble and the SUSAN detector were compared. It was found that, for lower detection probabilities, both detectors produced similar numbers of false alarms. For higher detection probabilities however, the Noble detector had a significantly lower false alarm rate.

2.4 The SIFT detector

SIFT stands for Scale Invariant Feature Transform, and was designed by Lowe [13] to address the need for detecting features which are partially invariant to affine and projective transformations.

The first step in the SIFT algorithm is to construct an image pyramid by successively convolving the image with Gaussians having a width of $\sqrt{2}$ pixels, followed by subsampling. The output of the pyramid is an array $\mathbf{B}_k$ which contains the smoothed images, and $\mathbf{A}_k$ which contains the difference between images smoothed at successive scales, and emphasises edge and corner features.

Once the image pyramids have been formed, the second SIFT step is detection, which basically looks for local maxima (or minima) in the pyramid $\mathbf{B}_k$. This step effectively detects blobs which are of higher or lower intensity than its immediate background, and also the approximate size of the blob (corresponding to the height of the maxima/minima in the image pyramid). Each of these detections is referred to by Lowe as a "feature key".

After detection, the orientation of each feature key is estimated. The orientation estimate uses the edge pyramid $\mathbf{A}_k$, to construct gradient magnitudes $\mathbf{M}$ and gradient orientations $\mathbf{R}$ at each level of the pyramid, as defined by

$$\begin{aligned} \mathbf{M}_k &= \sqrt{\left(\frac{\partial}{\partial x}\mathbf{A}_k\right)^2 + \left(\frac{\partial}{\partial y}\mathbf{A}_k\right)^2} \\ \mathbf{R}_k &= \arctan\left(\frac{\partial}{\partial y}\mathbf{A}_k / \frac{\partial}{\partial x}\mathbf{A}_k\right) \end{aligned}$$

For each feature key, and at each level, an Gaussian weighted ring of pixels about the central pixel (with a width three times the scale of the detection) is extracted. A histogram of the edge intensity of the pixels above one tenth of the maximum edge intensity is then constructed. The histogram has 36 bins covering the 360 degree range of rotations, and the maximum position is then defined as the orientation.

In order to facilitate matching between sets of SIFT detections from separate images, the local shape information from each feature key is stored. This is achieved using the previously computed $\mathbf{M}_k$ and $\mathbf{R}_k$ at the scale level of the detection to produce a set

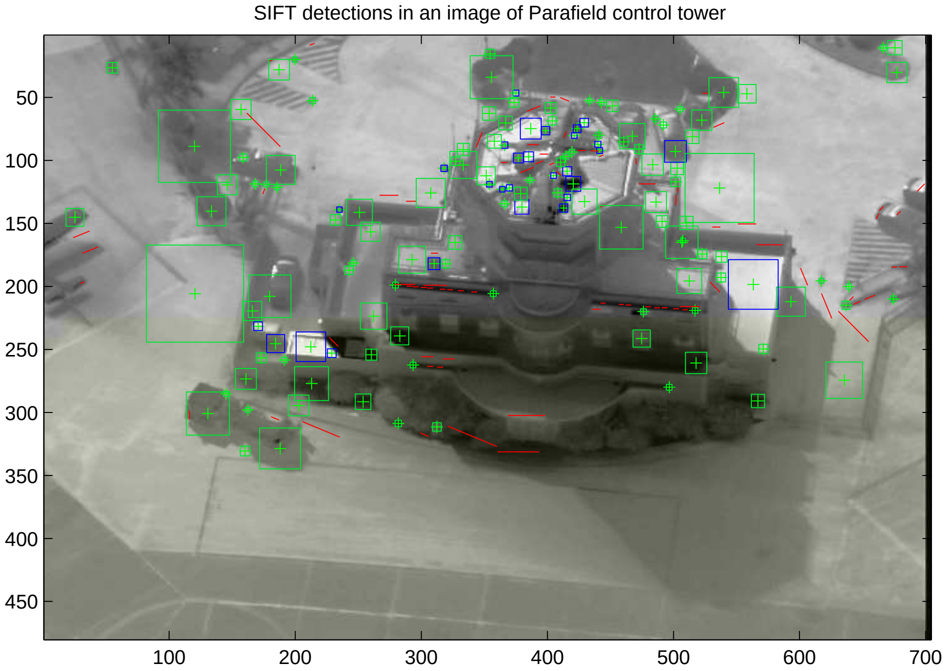


Figure 4: *SIFT detections in an example image*

of oriented edge images. Lowe splits the edge orientations into 8 evenly spaced bins, centered on the detection orientation, stores the gradient magnitude in each direction, then subsamples the resulting image to 20 points, which gives a total of 160 edge image points describing the shape information. By a direct comparison of these features between images, correspondences between SIFT detections may be made automatically.

Figure 4 shows the features detected using SIFT (as implemented by Scott Ettinger in MATLAB) in an image from the Parafield fly-over. As can be seen, few of the detections correspond to true corners, or in fact any part of the scene which would prove especially useful in reconstructing meaningful 3D models of the building. Also, there is no threshold to be set as all local maxima in the image pyramid space are classed as detections. This means that it is not possible to increase the density of points to be tracked, so that large areas of the building will not be accurately reproduced by any algorithm relying on these detections. There has been greater success however with using SIFT for structure from motion in office environments, where the sizes of the detected blobs are generally quite small compared with the size of the object to be modelled.

2.5 Local energy detector

The local energy method was based on a simulation of neural processing in a biological visual system by Heitger [7], which was then modified by Robbins and Owens [21]. Like the Shi-Tomasi detector, it searches for points at which the local intensity varies in perpendicular directions. The local energy implementation first applies an edge filter, which should detect linear features corresponding to step edges and narrow lines. A similar filter is then applied to find variations in the edge strength along the length of the edge. Detections are made separately over a set of edge orientations, and then combined.

Rather than use standard edge detectors, Heitger used a formulation based on the Gabor filter. Standard Gabor filters in one dimension take the form,

$$G_{even}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\cos(2\pi\nu_0 x), \qquad G_{odd}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\sin(2\pi\nu_0 x),$$

where ν_0 is the spatial frequency and σ is the overall scale. The two orthogonal functions are tuned for the detection of different characteristics. While the even function detects bright or dark pixels, the odd function detects step changes in intensity. As a result, any discontinuities in a signal $f(x)$, can be detected by locating peaks in $(G_{even} * f)^2(x) + (G_{odd} * f)^2(x)$, which is described as “local energy”.

The non-zero response of G_{even} to a constant input is undesirable for discontinuity detection, so the filters were modified to give the S-Gabor functions

$$\begin{aligned} G_{even}(x) &= \exp\left(-\frac{x^2}{2\sigma^2}\right)\cos\left\{2\pi\nu_0 x\left[k\exp\left(-\frac{\lambda x^2}{\sigma^2}\right)+1-k\right]\right\} \\ G_{odd}(x) &= \exp\left(-\frac{x^2}{2\sigma^2}\right)\sin\left\{2\pi\nu_0 x\left[k\exp\left(-\frac{\lambda x^2}{\sigma^2}\right)+1-k\right]\right\}, \end{aligned}$$

with $k = 1/2$ and λ the smallest root of $\int_{-\infty}^{\infty} G_{even}(x)dx = 0$. This modification reduces the frequency away from the origin. It is readily confirmed that roots depend only on $\nu_0\sigma$, and exist if that product exceeds approximately 0.38.

For a discrete signal, the filters can be constructed using the same formulas, and manipulated using the discrete Fourier transform (DFT) for a filter length equal to the signal length. The minimum value of $\nu_0\sigma$ and the value of λ for fixed $\nu_0\sigma$ are almost independent of σ for $\sigma \geq 2$ and long signals. The plots in both local energy papers [7, 21] resemble those for $\nu_0\sigma = 0.4$, so a value for σ determines values for ν_0 and λ.

Two-dimensional filters can be constructed using the McClellan transformation [16]. For an $m \times n$ image, the (i, j) element of the DFT (modulo (m, n)) refers to frequencies of i/m and j/n cycles per pixel (cpp). The combination is related to a radial frequency r/N cpp, where N is a suitable one-dimensional filter size such as $\max(m, n)$, through

$$\cos(\pi r/N) = \cos(\pi i/m)\cos(\pi j/n).$$

The design properties of a symmetric one-dimensional filter are then transferred to two dimensions by setting the two-dimensional DFT at (i, j) equal to the one-dimensional DFT at r, found by interpolation if necessary.

The symmetric two-dimensional filter can be made directional by multiplying the DFT by an angular term $\cos^{2k}(\varphi - \varphi_0)$ where $\tan\varphi = (j/n)/(i/m)$, φ_0 is the angle of orientation, and $k = 3$ is recommended [21]. The result is the DFT of the desired filter, to which the inverse DFT must be applied. Then a directional, two-dimensional version of G_{even} can be produced for any desired φ_0.

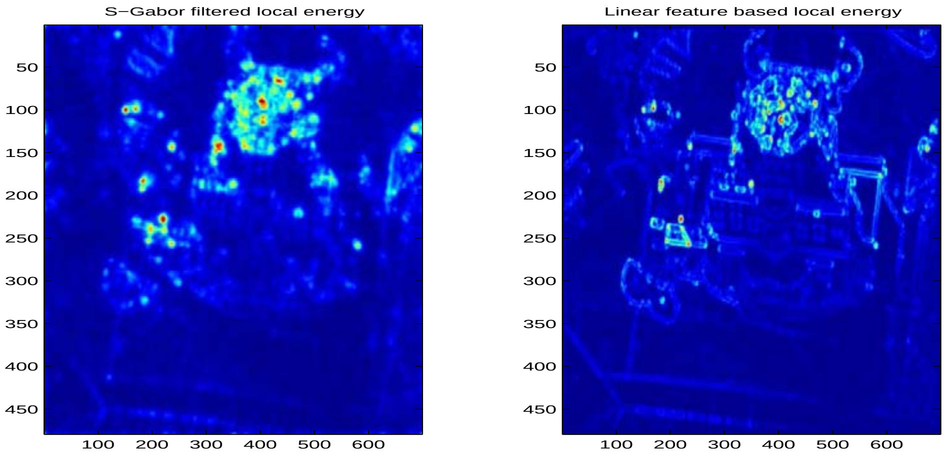
Both local energy papers [7, 21] claim to use the same method to produce a two-dimensional version of G_{odd}. The one-dimensional filter is odd and has an odd (and pure imaginary) DFT. The McClellan transform uses only the positive half of this DFT to construct a two-dimensional DFT, but the DFT should be pure imaginary and negated by a 180^o rotation if it is to produce an odd real directional filter. It is assumed here that the angular term for the odd directional filter is $\cos^{2k}(\varphi - \varphi_0)\ \mathrm{sgn}(\cos(\varphi - \varphi_0))$, so that the DFT is odd and resembles the DFT of G_{odd} along any line through the DC point. Then the two-dimensional versions of G_{even} and G_{odd} agree qualitatively with the spatial and Fourier domain figures in both papers.

For a two-dimensional signal $f(x, y)$, the function defined by

$$F(x, y, \varphi_0, \sigma) = \sqrt{(G_{even}(\varphi_0, \sigma) * f)^2(x, y) + (G_{odd}(\varphi_0, \sigma) * f)^2(x, y)}$$

detects linear features at orientations near φ_0 (zero for lines in the $\pm y$ direction) at scale σ. If the operation is applied a second time with an orthogonal orientation, the ”oriented two-dimensional local energy”

$$E(x, y, \varphi_0, \sigma) = \sqrt{(G_{even}(\varphi_0 + \frac{\pi}{2}, \sigma) * F)^2(x, y) + (G_{odd}(\varphi_0 + \frac{\pi}{2}, \sigma) * F)^2(x, y)}$$


S–Gabor filtered local energy

Linear feature based local energy

Figure 5: *Comparison of local energy maps using different edge filter templates*

detects lengthwise discontinuities in these features, and these are often places where linear features at orientations near φ_0 terminate or reach corners or junctions. The "two-dimensional local energy" is obtained by evaluating E for a range of orientations (Robbins & Owens [21] recommend ten equally spaced values of φ_0 over a 180^o range) and summing the results. Maxima in this local energy will correspond to corners, junctions and endings of linear features at any orientation, as well as strong point features or "blobs".

The S-Gabor filter template used for the edge detection was inspired by similarly shaped biological filters found in the vision system of a cat. No theoretical justification has been provided for using this particular filter, so alternative edge detectors might provide superior corner detection. Figure 5 shows a comparison of the local energy using the S-Gabor filter, and using a sharper linear filter. This linear filter was similarly designed to be a combination of orthogonal odd and even filters, each of which had zero mean. The relative intensity scaling of the filters was also chosen so that the sum of the squares of the even and odd components were both equal, which is a characteristic that the S-Gabor filter pairs also seem to have. This means that the filter should not show a preference for detecting step edges compared with lines. The two images in Figure 5 appear to have consistent local maxima, but the Gabor filter based image has much broader maxima, and seems to have more noise. The linear filter based method however shows a greater response to edges.

2.5.1 Modifications to the local energy

If true corners are to be located, the local energy method needs to be modified to suppress its responses to line endings and point features. A simple way to achieve this uses the fact that the oriented two-dimensional local energy $E(x, y, \varphi_0, \sigma)$ may be treated as a function of φ_0 with period π. A point feature will contribute to the energy for all orientations, and so to the DC term in the Fourier series for the energy. A line ending will contribute mainly to a single orientation and so mainly to the fundamental frequency. Two perpendicular line endings, corresponding to an ideal 90^o corner, will contribute mainly to the second harmonic. Therefore, to emphasise corners, the sum of the oriented energies is then replaced by the second harmonic amplitude

$$\sqrt{\left[\sum_{i=0}^{n-1}\cos\left(\frac{4\pi i}{n}\right)E\left(x,y,\frac{\pi i}{n},\sigma\right)\right]^2+\left[\sum_{i=0}^{n-1}\sin\left(\frac{4\pi i}{n}\right)E\left(x,y,\frac{\pi i}{n},\sigma\right)\right]^2}, \qquad (5)$$

where $n = 10$ is the setting already recommended above.

The autocovariance matrix used to define the Shi-Tomasi, Harris and Noble corner detectors, where the three unique matrix elements g_x, g_y and g_{xy} measure the edge strength in the x, y and $\pm 45^o$ directions. This means that the corner energy terms will be $g_x g_y$ and g_{xy}^2. Since the gradients are periodic over 2π rather than π, the second harmonic amplitude defined in equation (5) will be equivalent to $g_x g_y - g_{xy}^2 = det\mathbf{A}$, where $\mathbf{A}$ is the autocovariance matrix defined by

$$\mathbf{A} = \begin{bmatrix} g_x & g_{xy} \\ g_{xy} & g_y \end{bmatrix}.$$

This is the main term in the Harris detector (which is $det\mathbf{A} - \lambda\ tr\mathbf{A}$), although an extra term was required to further suppress edges.

The similarity between the local energy and the autocovariance methods also suggest that the Harris, Noble and Shi-Tomasi detectors can be reformulated in the local energy context. There are several ways in which this could be done, but the most obvious is to equate the eigenvalues of the autocovariance matrix with the maximum and minimum oriented local energies, so

$$\lambda_{1,2}(x,y) \equiv \left\{ \overset{max}{\varphi_0}\ F(x,y,\varphi_0,\sigma)\ ,\ \overset{min}{\varphi_0}\ F(x,y,\varphi_0,\sigma) \right\}.$$

Each of the autocovariance based detectors then becomes

$$\begin{aligned} \text{Shi-Tomasi:} \quad & \lambda_2 \\ \text{Noble:} \quad & \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2} \\ \text{Harris:} \quad & \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)^2 \\ \text{Intensity invariant Noble:} \quad & \frac{\lambda_1\lambda_2}{(\lambda_1+\lambda_2+\sigma^2)^2}. \end{aligned}$$

A comparison of the performance of these local energy based measures is provided in Subsection 2.7.

2.6 Hough transform methods

Hough transform methods relate detected features to parametric descriptions of objects to which those features belong. For example, an edge detection is related to the parameters (slope and intercept, say) of any straight line that passes through the point of detection. The detection "votes for" every compatible object, and a table of votes for all objects (the "Generalised Hough Transform" of the input image) is maintained as the detections are considered. Finally, objects with the most votes are considered to be detected. In the case of lines, this approach allows broken or partially obscured lines to be detected so long as the visible pieces are collinear.

This method can be applied to complex objects with many parameters to specify, but it then has the difficulty that a larger table of "votes" must be kept. At best, only a discrete set of values can be considered for each parameter, so a detected object is not identified exactly. Supplementary measurements on detected features can reduce the storage and computing requirements. Thus, if the orientation of an edge is estimated (even approximately), the set of relevant straight lines can be reduced.

Specialised HT methods for polygons of various degrees of symmetry, and corners, are discussed in Davies [3]. These are versatile enough to allow for rounding of corners by manufacturing imperfections, but impose restrictions like the size of angles in objects. They do, however, suggest a simple and novel approach for corners with general angles and for junctions of two or more lines.

The possibility of a straight line edge passing through 'A' needs supporting evidence from nearby edge pixels, which should have an orientation compatible with the edge passing through 'A'. For a corner to exist at 'A', there should be evidence of two or more edge segments intersecting at 'A', with an angle between two of them significantly different from 180^o. Only information a small distance from 'A' (say less than five times the pixel size) need to be considered when determining how likely the pixel represents a corner. One possible implementation of a Hough transform based corner detector based on this observation is as follows:

- **Edge detection:** Use a standard edge detector on the image, and threshold to give a set of edge pixels with their orientations.

- **Corner voting:** Each edge pixel 'B' provides a vote to neighbouring pixels 'A' which are within a certain distance of 'B', and are within some specified angle from the orientation of the edge through 'B'. Each vote is a vector of the form $k(1, \cos 2\theta, \sin 2\theta)$, where θ is the direction of 'B' from 'A', and k could either be constant, or proportional to the edge strength at 'B'.

- **Corner detection:** A corner is detected if the first component of the accumulated vote is above some user defined threshold. The corner strength is given by

$$D = 1 - \frac{\sum_i {P_i}^2}{N}$$

  where (N, P_1, P_2) is the accumulated vector for that pixel.

The first edge detection step is not necessary for the case when the size of the vote is proportional to the edge strength, because non-edges would have a relatively small contribution. It does however reduce the total amount of computation required for generating the Hough transform.

The second step accumulates the vector $(1, \cos 2\theta, \sin 2\theta)$ at each pixel 'A' in the image. The first variable indicates the total number of local edge points passing through 'A', but this will be large, not just at corners, but also at edges. The last two variables reduce the weighting on the edges. Consider the particular example of a straight edge with orientation θ_0 passing through 'A'. Then if 'B' is a pixel on this edge, it would contribute $(\cos 2\theta_0, \sin 2\theta_0)$. This contribution will be unchanged when 'B' is moved to the opposite side of 'A'. This means that if 'A' lies on an edge instead of a corner, the average accumulation $< \cos 2\theta_0, \sin 2\theta_0 >$ should lie on the unit circle, and so the corner strength $D = 0$. If, at the other extreme, the point 'A' is at the intersection of two perpendicular edges of equal strength, the mean of P will vanish and D achieves its maximum value of 1.

2.7 Evaluation of corner detectors

There have been a number of different methods for assessing the detection of corners. The measures suggested by Schmid *et al.* [24] and Tissainayagam and Suter [28] are based on the idea of repeatability of corner measurements. In the case of Schmid *et al.*, the performance measure was the percentage of points consistently detected in two images differing by an affine transformation. Tissainayagam and Suter however measure the consistency of detections in a static sequence. The two papers do not consider any corner detectors in common. In the first paper, the Harris detector gave higher repeatability under affine transformations than detectors by Heitger, Cottier, Horaud, Heitger and Förstner. In the second paper, the Noble and Shi-Tomasi detectors gave the most consistent results in static videos, compared with SUSAN and the Kitchen-Rosenfeld detectors.

There are a number of problems with solely using repeatability measures as a measure of performance in the the current structure from motion work. Firstly, there is no test whatsoever that the detected points correspond to corners. All that can be said is that the points (which might be the centres of blobs, as in SIFT, or local intensity maxima) are detected consistently. For buildings, real corner points are more likely to correspond to points that are useful in creating a wire-frame model (*i.e.* the junctions of three surfaces), and other points of interest are of less use, regardless of how consistently they are tracked. Secondly, the performance measure also implicitly assumes that features will be tracked between frames by detecting a fixed number of corners in each frame, and then associating the detections. This may not be the ideal tracking method. For instance, suppose an almost perfect corner detector existed which consistently produced a value close to 1 at corners, with 0 elsewhere. If N corners are detected in each frame, and the actual number of corners is $N_c > N$, then because there is no dominant corner, in each frame there will be a probability of N/N_c of losing track of a point. This means that the ideal corner detector could appear to perform very poorly based on this measure of repeatability. Some better methods for tracking are described in Section 3.

Rockett [22] and Martinez-Fonte *et al.* [15] have used a more empirical method for assessing corner detectors. In these papers, examples of true corners and non-corners are provided to each of the detectors. For each threshold level of the detector, the corner detection probability and the false alarm rate are measured, and plotted as an ROC curve. The area under the ROC curve may then be used as a measure of performance of the corner detector. Rockett used simulated corners in his experiments, which allowed him to obtain accurate results using large data sets. Martinez-Fonte *et al.* used manually selected points from real images of buildings. Again, there were no corner detectors in common between the studies. Rockett found that the Harris detector gave the best detection rate (although it did not localise well) when compared to the Paler and Kitchen-Rosenfeld detectors. Martinez-Fonte *et al.* found that the Noble detector gave marginally better results than SUSAN.

For this report, the performances of a wide variety of corner detectors have been compared, following the work of Martinez-Fonte *et al.* [15]. Here, a set of 92 structurally important corner points were manually selected in the first frame of the Parafield fly-over sequence, as shown in Figure 6. These were then automatically tracked, using local correlation matching (see Section 3.2), throughout the sequence of 51 images, each of 704 × 480 pixels. The corner detectors were then applied to each image, the detected points close to stronger detections were suppressed, and then the remaining detections were compared with the ground-truth points to generate ROC curves.

Figure 7 shows ROC curves for seven commonly used corner detectors. It should be noted that the SUSAN detector uses only the initial measure of cornerness, and does not

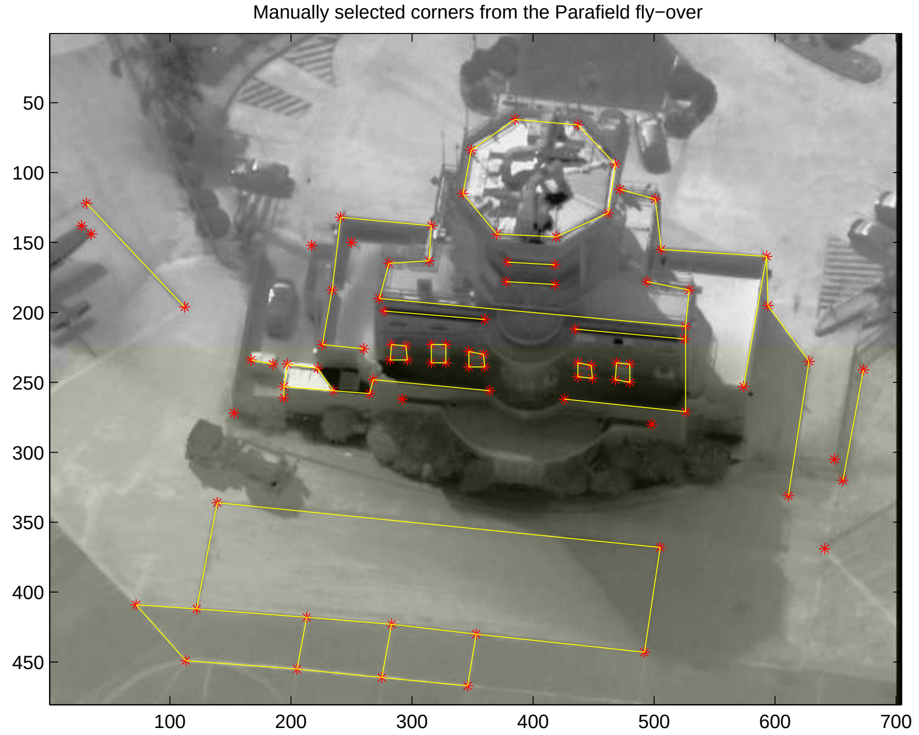


Figure 6: *Manually selected corner points from the first image frame*

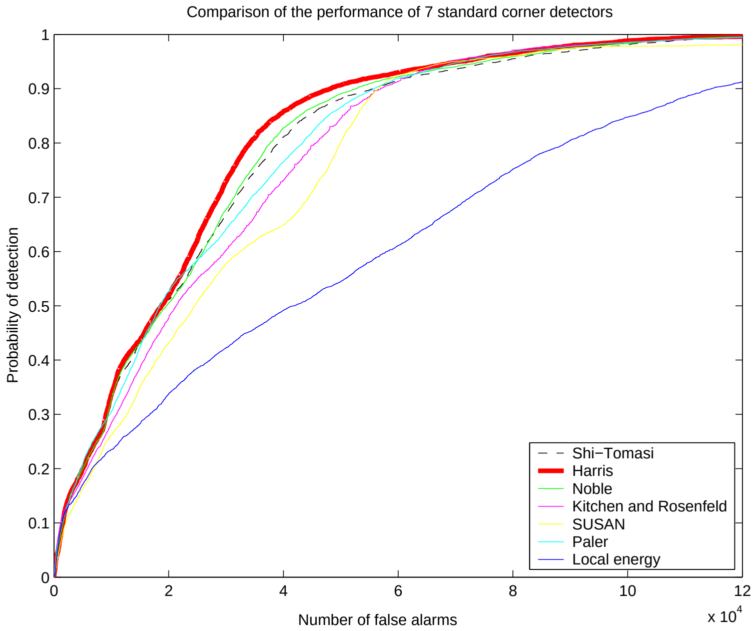
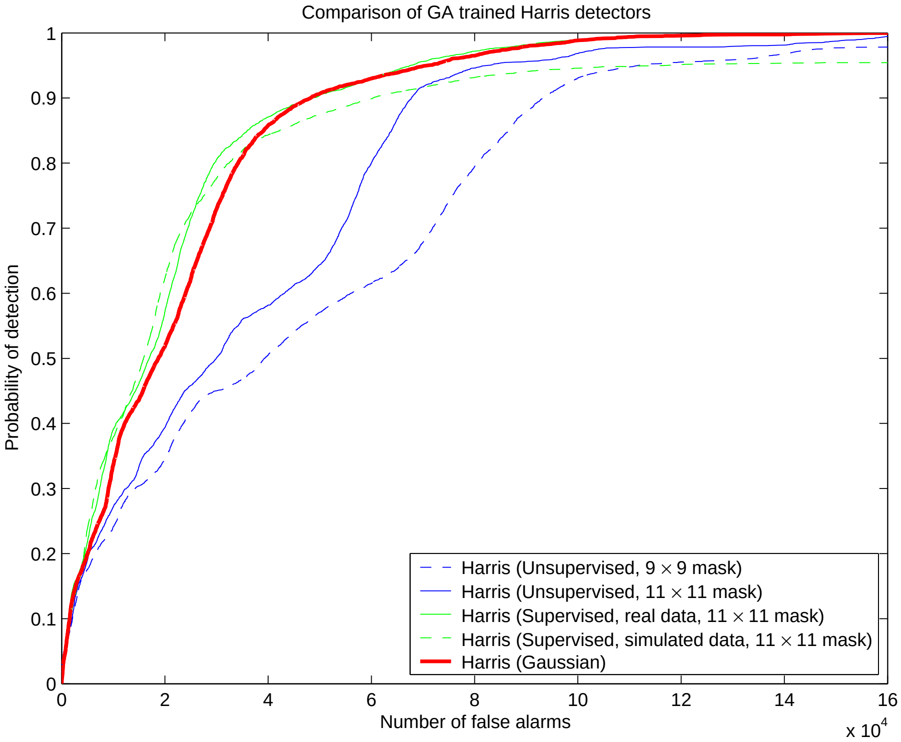


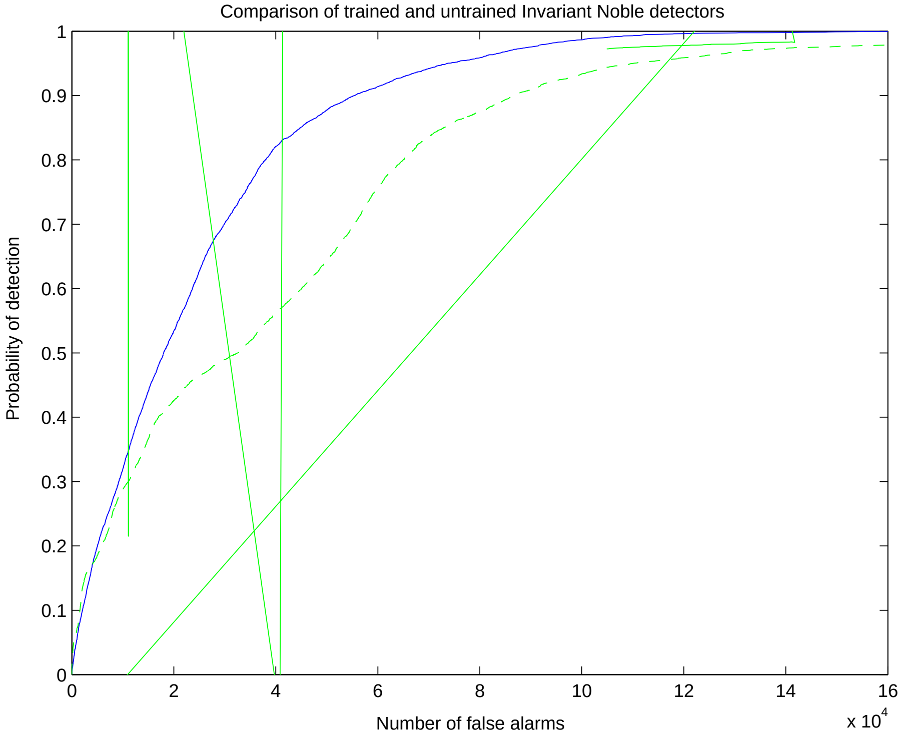
Figure 7: *Comparison of performances of seven types of corner detector*

include any of the post-processing operations for reducing the number of false alarms. The effect of this processing will be considered later. The Paler detector [20] is based on the observation that a median filter tends to keep edges intact, but blur the corners. The measure of cornerness is then just a difference between the pixel intensity and the intensity of the median of the local neighbourhood of pixels. The local energy detector is the standard implementation of Robbins and Owens [21], which uses sequential S-Gabor based edge detection (see Section 2.5). The remaining detectors as as described in the previous subsections.

From the ROC curves, the Harris detector appears to be the best performing, consistently giving either fewer or equal numbers of false alarms for a specified detection probability. This result further supports the evaluations of previous studies (Schmid *et al.* [24] and Rockett [22]) which concluded that the Harris detector performed better (if we ignore corner localisation) than competing detectors. In Subsection 2.1, it was proposed that the Harris corner detector could be improved by using a weighting function different from a Gaussian in implementing the detector. This function could be determined using a genetic algorithm with some reward function based on a corner training set. Figure 8 shows some ROC curves obtained using this method to maximise a number of different performance metrics.

The first set of ROC curves from Figure 8 show implementations of the Harris detector using weighting functions optimised separately for three different performance metrics. The first metric is the repeatability metric used by Tissainayagam and Suter. Here, the corner detector is first applied to a frame of the sequence to give 100 detections. Then, white noise is added to the image, and the process repeated. The average number of detections repeated is then used as a measure of the performance of the detector. This


Comparison of GA trained Harris detectors
Probability of detection
Number of false alarms
x 10^4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
2
4
6
8
10
12
14
16
Harris (Unsupervised, 9 × 9 mask)
Harris (Unsupervised, 11 × 11 mask)
Harris (Supervised, real data, 11 × 11 mask)
Harris (Supervised, simulated data, 11 × 11 mask)
Harris (Gaussian)


Comparison of trained and untrained Invariant Noble detectors
Probability of detection
Number of false alarms
x 10^4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
2
4
6
8
10
12
14
16

training method has been termed unsupervised, since no information about the real corners is available during training. It can be seen that the ROC curves for the unsupervised case are much worse than if the default Gaussian weighting had been used. This is because the detector is becoming better trained to detect regions of high intensity and small blobs, which can be consistently detected, but are not the corner points which are of most interest to detect.

The second metric used for training the Harris detector, the performance of which is shown in Figure 8 (labelled as Supervised, with simulated data) is that of Rockett [22]. This metric is based on an artificial but realistic looking training set containing corners and non-corners. The set of corners was simulated with uniformly distributed corner angle and orientation (with the minimum angle required for a corner set to be 10^o). The set of non-corners consisted only of straight lines and edges of arbitrary orientation. Both types of images had additive white noise applied. The area under the ROC curve for the resulting simulated data was then used to train the genetic algorithm for the weighting function. Figure 8 shows a noticeable improvement in performance for the first 80 percent of targets, however it is much worse above this. From this, it seems likely that the simulated model used for the corners differs from the actual corners to be detected in about 20% of cases.

The final metric used in training the Harris detector with modified weighting function is based on the empirical area under the ROC curve, and is referred to in Figure 8 as supervised with real data. To avoid potential problems with overfitting, only the ground-truth and false alarms from the first image frame were used for constructing the ROC curves used for training the GA. As shown in Figure 8, using the area under that ROC curve as the reward criterion resulted in an overall improvement in the detection performance.

The second diagram of Figure 8 repeats some of the experiments from the previous diagram for the intensity invariant Noble detector, which also uses a Gaussian weighting distribution by default. As for the Harris detector, using Tissainayagam and Suter's repeatability for training reduced the overall ability to detect corners. The reduction was not quite as marked as for the Harris detector because of the impossibility of training the intensity invariant detector to detect bright patches at the expense of corners. Again, training on the area under the ROC improved the detection performance, although this was much more significant for an 11×11 window size than for a 9×9 window. Increasing the size further to 13×13 however appeared to make little difference, and is not shown here.

In Figure 7, the local energy detector appeared to perform very poorly. This, at least partially, will be due to the extremely broad maxima in the local energy around each detection, which results in multiple detections for a given peak. Non-maximum suppression may significantly improve this performance, although Rockett [22] found that this actually harmed performance when applied to the Kitchen-Rosenfeld detector. There are, however, a number of other methods for improving the local energy detector which have not yet been tested.

Subsection 2.5 described in detail the operation of the local energy detector. Some of the different possible implementations of the detector included changing the type of edge filter to a pair of orthogonal linear edge filters (see Figure 5 in Subsection 2.5), and using the second harmonic of the 2D local energies as defined by equation (5). There are also different ways to apply the perpendicular local energy filters: sequentially, or separately

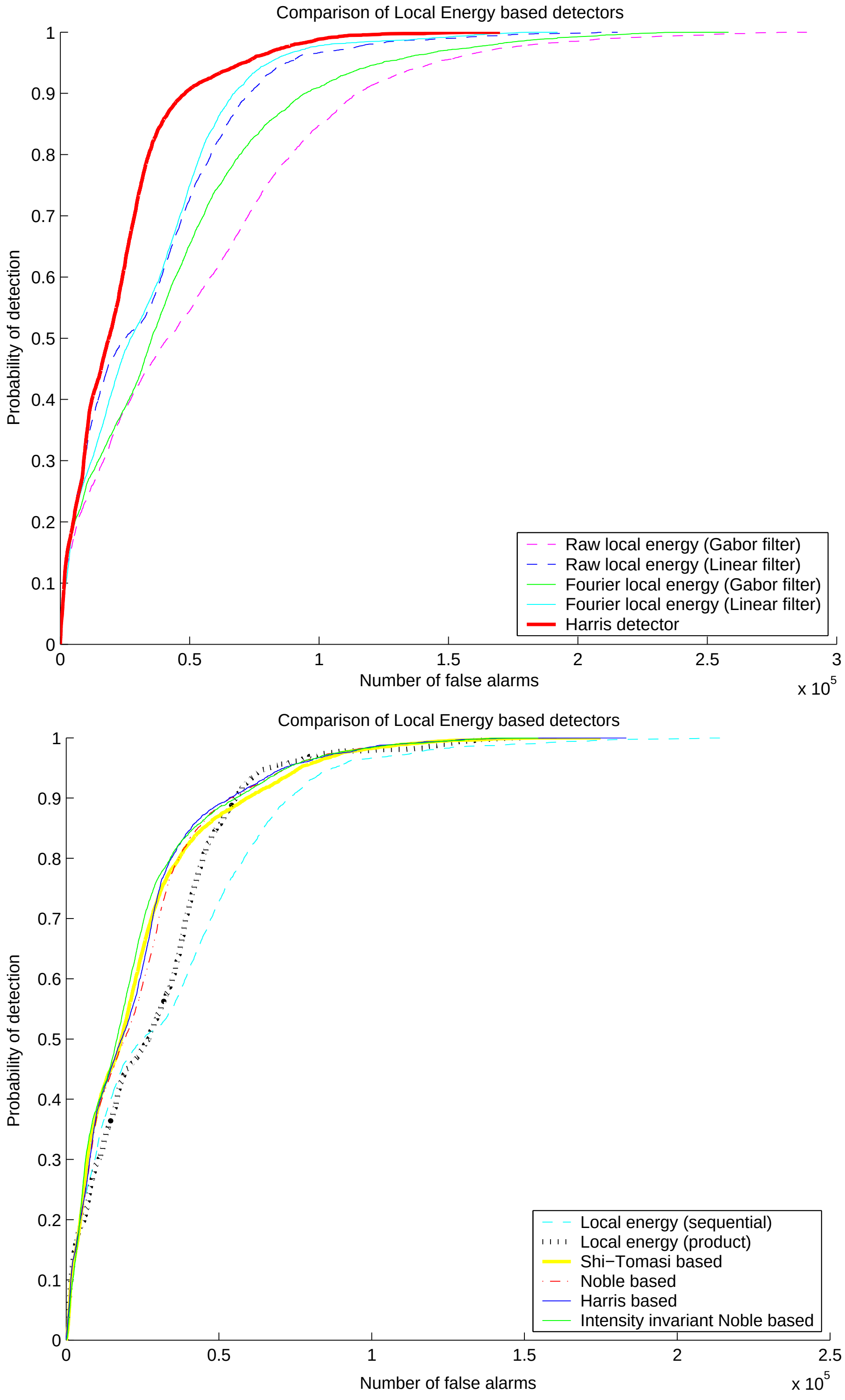


Figure 9: *Comparison local energy detector performances*

with the results combined later. Numerous methods also exist for combining the edge energies, which do not merge perpendicular edge orientations. Some of these methods are inspired by the autocovariance matrix based detectors (the Harris, Noble and Shi-Tomasi detectors). Figure 9 gives a comparison of a number of corner detection results for local energy implementations.

The first set of ROC curves clearly shows the effect of the template type. The two curves obtained using the S-Gabor edge filter are much worse than the equivalent curves obtained using the orthogonal linear filters. In Heitger's paper [7], it was implicitly assumed that the S-Gabor was best, because it appeared in a biological system which had evolved for the task. However, biological systems have great difficulty approximating step edges, and it may be that in the final design that there was a trade-off between the edge filter shape and the sensor packing density. In this application, no such trade-off is necessary, and it appears that the edge filter shape could easily be improved. Similarly, the two curves which use the second harmonic component of the local energy, as defined by equation (5) give better results than for the raw local energy. For the S-Gabor filter case, the improvement is significant, but is only marginal for the linear filter case.

The second set of ROC curves all use the linear filter. The original formulation of the 2D local energy for a given orientation φ applied the edge filter $G(\varphi_0, \sigma)$ to the image, and then followed up with the application of the perpendicular filter $G(\varphi_0 + \pi/2, \sigma)$. One can obtain a similarly behaving detector by defining

$$\tilde{E}(x, y, \varphi_0, \sigma) = \sqrt{F(x, y, \varphi_0, \sigma)F(x, y, \varphi_0 + \pi/2, \sigma)}.$$

This product detector (as distinct from the original sequential detector) should suffer from less blurring. As can be seen from the ROC curve, it also substantially improves the detection performance on real corners. The next few curves were obtained by taking λ_1 and λ_2 as the maximum and minimum edge energies for each point in the image over

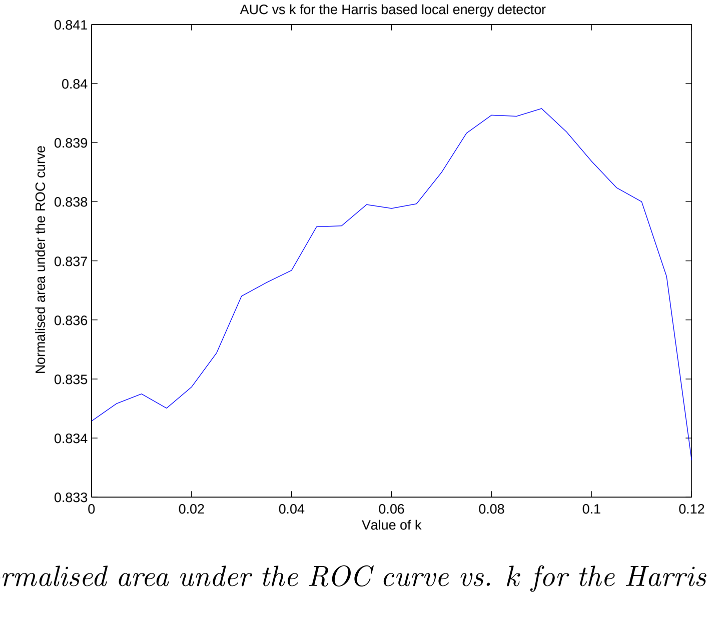


Figure 10: *Normalised area under the ROC curve vs. k for the Harris based local energy detector.*

the various orientations. As defined at the end of Section 2.5, these are then combined in various ways to give different detectors. For the Harris based detector and the intensity invariant Noble detector, the unknown parameters were chosen to be $k = 0.085$ and $\sigma = 1000$. These values were chosen by optimising the performance on the first image in the sequence, which was effectively used as training data. Figure 10 shows a plot of the normalised area under the ROC curve as a function of the parameter k. All of the different combinations proved more effective than just using products of perpendicular edge filter responses, but there was not a lot of difference between the various combinations.

The generalised Hough transform was described in Subsection 2.6. It is based on accumulating evidence of corners at each point based on edge information from nearby points. Only edges with directions within some user defined angle θ_{err} of passing through a pixel contributes to the evidence at that pixel. There are two related corner measures which are produced by the method. The first is the combined edge strength K of all of the neighbouring pixels contributing to a corner. This measure will be large for edges as well as corners. The second measure D is a quantity between 0 and 1, which relates to the angle subtended by the corner. This effectively eliminates the response due to edges, but may still be large in cases where there is not enough evidence.

Figure 11 shows the performance of a number of implementations of the generalised Hough transform detector. The first two curves combines the edge strength K and cornerness measure D into a single measure by multiplying the two. The performance of this product, for the Parafield data, was compared for two different values of θ_{err}. It was

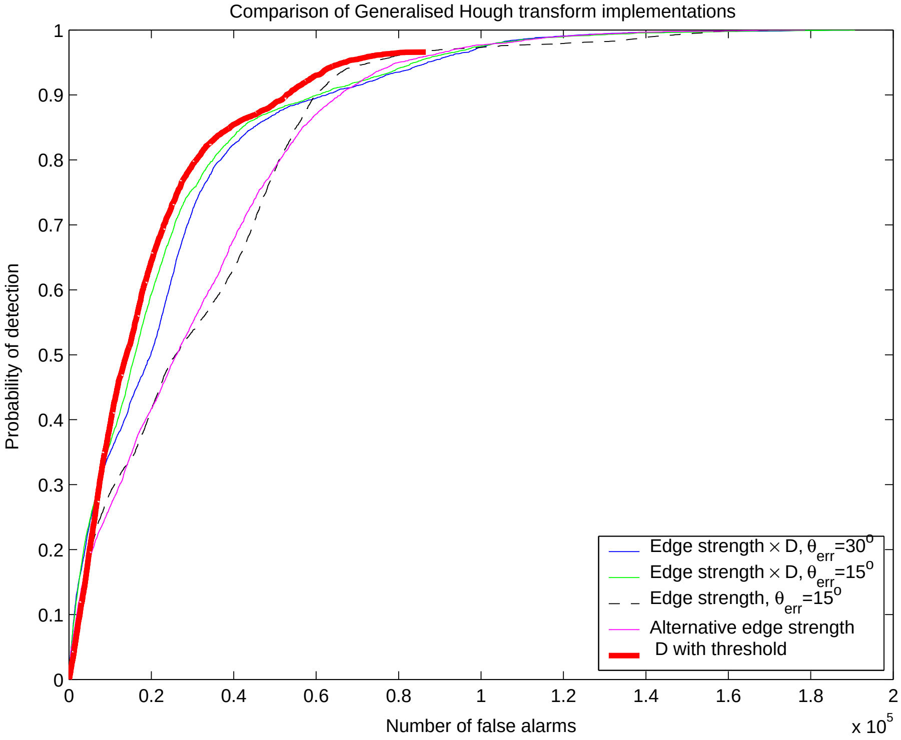


Figure 11: *Comparison of implementations of the generalised Hough transform detector*

found that $\theta_{err} = 15^o$ gave a noticeable improvement over $\theta_{err} = 30^o$. Further simulations (not displayed here) found that decreasing θ_{err} still further did not result in any more improvement.

The next two curves of Figure 11 examine the effect of using the edge strength K by itself. The second of these curves used an alternative method for determining which pixels contributed to the edge strength. Here, the edge direction of the candidate corner pixel was determined, and any local pixels within 3 pixels of this estimated edge were automatically ignored. The remaining pixels could contribute to the edge strength as discussed previously. The aim of this modification was to try to remove the contribution of edges in a different way from the one chosen in computing the value of D. As can be seen from the ROC curve however, this alternative method shows no obvious advantage over the nominal method for calculating K. The final ROC curve of the figure combines K and D by thresholding based on K (where the threshold was chosen to give a PD of 96%) and then sorting the corners based solely on the corner measure D. This method gave the best result of all of the GHT based detectors.

3 Point trackers and related detectors

In order to use most of the standard shape from motion algorithms, the position of features needs to be known throughout a number of frames, and so requires some sort of tracking. This section describes a number of simple and commonly used tracking algorithms for point features. Each point tracking method can also be associated with a point detection algorithm using the principle expounded by Shi and Tomasi [25], that the best features to track are generally the ones that are easiest to track. This section therefore also analyses a number of new corner measures relating to the ease with which a point can be tracked to the next frame for any given tracking method.

Several types of tracking methods have been considered. In the first type, sets of features are detected in each frame, and the tracking algorithm consists of finding associations between the detected points. A brief description of some of this nearest neighbour data association is provided in Subsection 3.1. The second type of tracking detects features in the first frame only, and then determines their motion between frames. The KLT tracker, perhaps the most commonly used feature tracker, is of this second type, and is discussed in Subsection 3.2. Correlation matching, described in Subsection 3.3 is also of this type.

Every tracking algorithm has an implicit model describing how the features move. Many of them use the fact that in a video sequence, consecutive frames are temporally close together, so the feature is unlikely to have moved far. In this case, the predicted position in the next frame is just the position in the current frame. This prediction can be improved upon by estimating velocities and accelerations for each point, or by using correlation properties between the motions of features in the same sets of frames. Subsection 3.4 describes a number of methods for improving the predicted positions of tracked features. The final subsection describes ways for determining whether a tracking method has jumped to the wrong feature, so that the track may be terminated.

3.1 Nearest neighbour data association

Unlike some of the tracking algorithms discussed later (such as the KLT tracker and correlation matching algorithm), Nearest Neighbour Data Association does not directly deal with the image. Instead, it is assumed that some interest point detector has been applied to each of the images to give a set of detections for each frame. NNDA then associates the detections in the new frame to those in previous frames, using the following simple algorithm:

- **Predict corner positions:** Using the positions of the detections in previous frames, predict where they are likely to appear in the current frame. A number of ways to do this are discussed in Subsection 3.4.
- **Data association:** Find the spatially closest detection in the current frame to each predicted position, and associate the two.

The performance of this algorithm will depend strongly on the reliability of the particular corner detector used to generate feature detections.

3.1.1 Nearest neighbour corner detection

In this subsection, a corner detector is described which measures the ease of tracking a feature using the NNDA tracking method. The specific metric used to describe this is the amount of image noise required to produce a certain likelihood of a mismatch in the tracker. The operation of the NNDA tracker is strongly dependent on the particular corner detector being used to find features within each frame prior to association. This means a different measure will be obtained when a different corner detector is used. In fact, an NNDA tracking performance measure could be computed based on a corner detector which was another NNDA tracking performance measure. At this stage, it is not certain what the effect of this recursion of the method might have. Since non-zero values will only occur at local maxima in the original corner map, further iteration may not add any performance to the resulting detector. This has not been simulated, however, due to the extreme computational requirements of such a task. In fact, even the first application of the method is fairly difficult, due to the analytic intractability of most of the common corner detectors.

Figure 12 shows a ROC curve based on the NNDA performance measure for the Shi-Tomasi detector. Due to the complexity of the detector, an analytical expression for the measure was not attempted. Instead, it was estimated using a Monte-Carlo method as follows:

- **Corner map:** Apply the Shi-Tomasi detector to the pristine image.
- **Loop:** For each of the local maxima in the corner map image, find the amount of noise σ which gives a 10^{-4} probability of a mismatch using NNDA. This is done by first initialising σ to, say, 1. Then applying the following:

  - **Store histogram:** Around each detection, extract a small square of imagery and apply 500 instances of noise with variance σ^2 to it. For each of these instances, apply the Shi-Tomasi corner detector and store the difference between the central corner value and the maximum of the neighbouring peaks. When this difference is greater than zero, the NNDA tracker will produce a false match.
  - **Determine false match probability:** It is not possible to directly measure probabilities of around 10^{-4} accurately with only 500 points. To increase the number to 10^5 however would result in an enormous increase in computation time. Instead, the tail of the distribution is estimated as decreasing with a power law. The parameters of this tail can be estimated using the Hill's estimator [8]. From this model, the probability of false alarm may be more accurately estimated.

- **Update noise variance:** Because the false match probability will be monotonically increasing with σ, a binary search method may be used to find the value giving the required false match probability of 10^{-4}. Here, a lower and an upper bound should be stored and the search interval halved, each time through the loop. Due to statistical fluctuations in the probability estimate, this may result in significant errors in the resulting noise estimate, but for this report, no fast alternative immediately suggested itself.

To reduce the large amount of computation required, the result in Figure 12 was obtained for only the first frame of the video, and the false alarm rate was assumed to be the same for the rest. It can be seen that the NNDA measure is very similar to the original

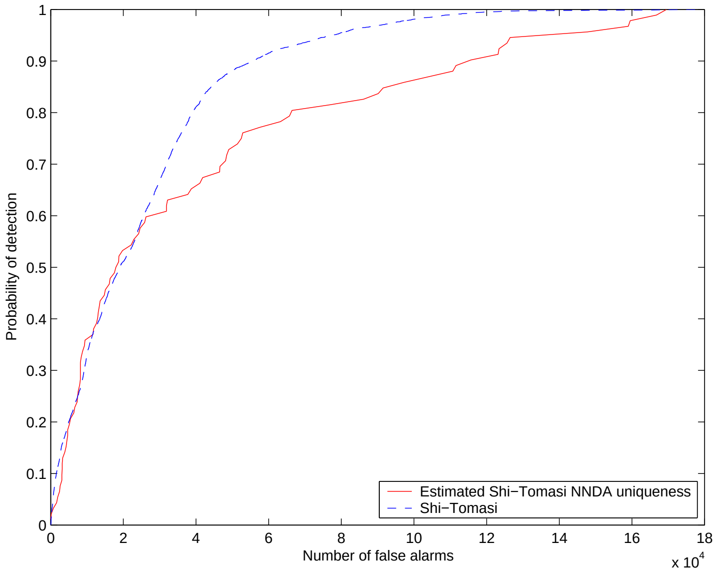


Figure 12: *ROC curves comparing the corner detection performances of the Shi-Tomasi detector and the detector based on the noise required to give a* 1 *in* $10,000$ *probability of a false match using the NNDA tracker.*

corner measure result up to a corner detection rate of about 0.6. This indicates that the statistics of detection are similar whether the noise is applied to the input image or to the corner map. For higher detection rates, it seems that the original Shi-Tomasi detector did not have local maxima corresponding to the remaining corners. Since the NNDA measure is only non-zero where there is a maximum in the original corner detection image, the new measure becomes noticeably worse.

3.2 The KLT tracker

The KLT (Kanade-Lucas-Tomasi) tracker is based on an image registration algorithm, first defined by Lucas and Kanade [14]. In this paper, whole images $F(\mathbf{x})$ and $G(\mathbf{x})$, differing by a translation were matched by minimising the error function

$$\int\int (F(\mathbf{x}) - G(\mathbf{x} - \mathbf{d}))^2 dxdy \tag{6}$$

with respect to the image displacement $\mathbf{d}$. This was solved by assuming that the displacement is small, and using a first order Taylor approximation for the image intensity to give

$$\begin{aligned} \text{Error} &= \int\int (F(\mathbf{x}) - G(\mathbf{x}) + \nabla G(\mathbf{x}).\mathbf{d})^2 dxdy \\ &= \int\int (F(\mathbf{x}) - G(\mathbf{x}))^2 dxdy + \left[2\int\int (F(\mathbf{x}) - G(\mathbf{x})\nabla G(\mathbf{x}) dxdy\right].\mathbf{d} \\ &\quad + \mathbf{d}^T \left[\int\int \nabla G(\mathbf{x}) \nabla^T G(\mathbf{x}) dxdy\right] \mathbf{d} \end{aligned}$$

This quadratic equation in the image displacement $\mathbf{d}$ will have an easily determined unique solution, given by the solution to the linear equation

$$\left[\int\int \nabla G(\mathbf{x}) \nabla^T G(\mathbf{x}) dxdy\right] \mathbf{d} = \left[2\int\int (G(\mathbf{x}) - F(\mathbf{x})\nabla G(\mathbf{x}) dxdy\right] \tag{7}$$

where the matrix on the left hand side is referred to as the autocovariance matrix $\mathbf{A}$ of the gradient, over a square window.

In order to remove any inaccuracies due to the higher order terms in the Taylor expansion, the procedure was iterated, until the estimate for the image displacement changed by less than one hundredth of a pixel. The paper also mentioned that the method could be extended to more general affine transformations, but no specific details were provided.

The Lucas and Kanade registration algorithm was first applied specifically to the problem of tracking feature points in a report by Tomasi and Kanade [30] in 1991. Here, a set of feature points are detected in the first frame and a small image chip is extracted and registered to the subsequent frames using the above registration algorithm. If the feature was correctly tracked, then the intensity error (or dissimilarity) given by equation (6) should be small. If the dissimilarity exceeded a certain threshold, it was assumed that track of the feature had been lost.

It was recommended, by Tomasi and Kanade, that the set of features in the first frame be chosen so that they are easiest to track. It was argued the solution would be least sensitive to noise when the matrix $\mathbf{A}$ associated with equation (7) was least singular, or when the minimum eigenvalue λ_2 was largest. The resulting interest point detector has come to be known as the Shi-Tomasi detector (although it should perhaps be more properly known as the Kanade-Tomasi detector), which was discussed more fully in Section 2.2. The complete system, consisting of the initial Shi-Tomasi detector, followed by translation based registration, and track termination based on dissimilarity, has become known as the KLT tracker.

Shi and Tomasi [25], published a more publicly available account of the KLT tracker in 1994, and as a result it has become more frequently cited than the earlier report. The main addition described in this paper is a full description of an extension of the registration step to a general affine transformation. According to the paper, this allows features to be tracked accurately for greater lengths of time as the camera view rotates or zooms.

The KLT algorithm has been tested against competing algorithms in a number of settings. For instance, Barron *et al.* [1] compares algorithms for solving the optical flow problem, which is to find the 2D projection of the 3D motion field of a scene, based on two images of the scene closely separated in time. This is implemented for the KLT by applying the registration algorithm to each point in the image, as if it were a feature being tracked. The performance of the algorithm was tested on various types of simulated imagery for which the true optical flow was known. Of the nine methods tested, by far the two best methods were the KLT method, and another by Fleet and Jepson [4], based on local phase information.

3.2.1 Another KLT based corner detector

The Shi-Tomasi detector, described in the previous section, defines corners as being related to the speed of convergence of the KLT tracker at any given point. This is not necessarily a useful measure of the ability of the KLT tracker to track, so an alternative has been considered. In this subsection, an expression is derived for the amount of image noise required to cause the KLT tracker to fail to track a specified point. The performance of this metric as a corner detector is also examined.

Suppose an image of pixel intensities $g_{i,j}$ is being tracked to a new noisy image $g_{i,j} + \sigma\varepsilon_{i,j}$, where $\varepsilon_{i,j}$ is a random variable with zero mean and unit variance. The KLT tracker will choose the translation (τ_x, τ_y) to minimise the error metric

$$\sum_{i,j} (g_{i,j} + \sigma\varepsilon_{i,j} - g_{i+\tau_x, j+\tau_y})^2.$$

To simplify calculations, it will be assumed that measures of the error metric at two different translations are statistically independent. For a given translation, the metric will be a sum of i.i.d. random variables, and so can be approximated by a Gaussian, regardless of the actual distribution of the pixel noise. The mean will be

$$\mu_{\tau_x,\tau_y} = \sum_{i,j}(g_{i,j} - g_{i+\tau_x,j+\tau_y})^2 + N^2\sigma^2$$

where N^2 is the number of pixels contributing to the error metric, and the variance can be shown to be

$$\sigma^2_{\tau_x,\tau_y} = 2N^2\sigma^4 + 4\sum_{i,j}(g_{i,j} - g_{i+\tau_x,j+\tau_y})^2\sigma^2.$$

The probability that the KLT error metric will be smaller for one of the off-centre pixels, and so will give an incorrect match, will therefore be

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{\tau_x,\tau_y}} \exp\left(-\frac{x^2}{2N^2\sigma^2}\right)\left[1 - \prod_{\tau_i,\tau_j}\left\{1 - erf\left(\frac{x + N^2\sigma^2 - \mu_{\tau_x,\tau_y}}{\sqrt{2}\sigma^2_{\tau_x,\tau_y}}\right)\right\}\right] dx.$$

This will be a monotonically increasing function of σ, which means that the amount of noise required to give a specified failure probability can be determined using a binary search method. Since this method must be repeated for every pixel in an image, it is computationally infeasible to compute the integral accurately. In the results described in this report, a 10 point trapezoidal rule has been used as an approximation, with the points chosen to be equally spaced over the interval $x \in [0, \min_{(\tau_x,\tau_y)\neq(0,0)} \sum_{i,j}(g_{x,y} - g_{x+\tau_x,y+\tau_y})]$. Figure 13 shows an example of a map of this measure for the first image in the Parafield fly-over. The corresponding ROC curve shows that the detector still does not appear to

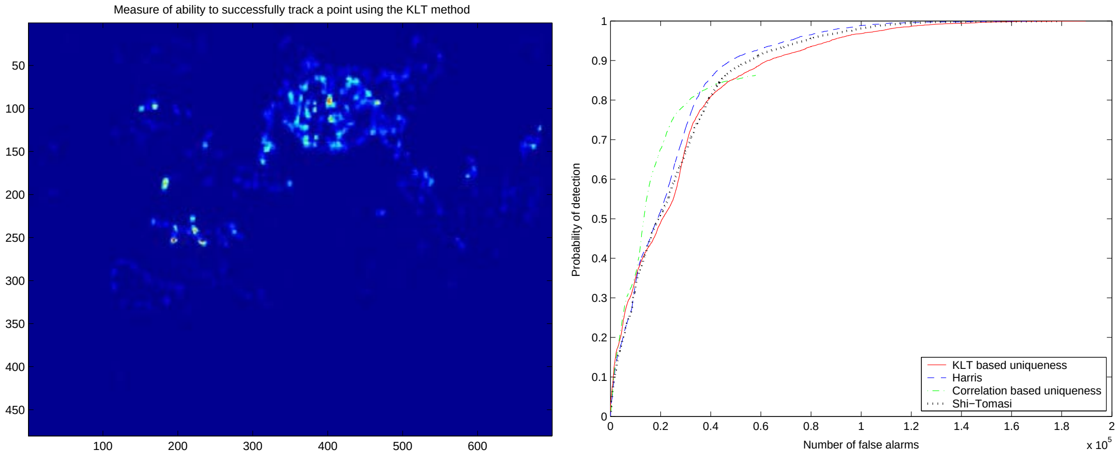


Figure 13: *Plot of a) the noise required to give a* 1 *in* 10,000 *probability of a false match using the KLT tracker, b) ROC curves comparing the performances of this detector and others*

perform as well as the Harris detector but, apart from at the highest detection probabilities, it still performs comparably with some of the better detectors. As shown, it seems to perform similarly to the Shi-Tomasi detector, which was also a measure of KLT tracker performance. Possibly the slightly lower performance of the uniqueness based measure is due to the low accuracy of the approximation to the above integral used in the simulations.

A corner measure derived similarly to the above KLT performance measure has also been considered by Kaneko and Hori [10]. They report very good detector performance, although their results have not been replicated for this report.

3.3 Correlation matching and other image measures

The KLT tracker, described in the previous subsection, can be used to follow a point by searching a small neighbourhood around its predicted position in the next image. A match is then made to the position which looks most similar to the image in the preceding frame. The distinguishing feature of the KLT tracker is the choice of dissimilarity measure, which for two images $g_{i,j}$ and $h_{i,j}$ is given by

$$C_1 = \sum_{i,j} (g_{i,j} - h_{i,j})^2.$$

This, however, is not the only plausible measure for the difference between two images. Smith *et al.* [26] provides a list of a number of such measures which have been frequently used in the literature. According to them, the most commonly used measure is the “standard cross-correlation”, given by

$$C_2 = \frac{\sum_{i,j} g_{i,j} h_{i,j}}{\sqrt{\sum_{i,j} g_{i,j}^2 \sum_{i,j} h_{i,j}^2}}. \tag{8}$$

This differs from the unnormalised version of this formula (which is just the numerator) that will be used in Subsection 3.3.1 to obtain a measure of uniqueness based on the difficulty of tracking each point. The other similarity measures collected by Smith *et al.* [26] are

$$\begin{aligned}
C_3 &= \frac{\sum_{i,j}(g_{i,j} - \bar{g})(h_{i,j} - \bar{h})}{\sqrt{(\sum_{i,j} g_{i,j}^2 - n\bar{g}^2)(\sum_{i,j} h_{i,j}^2 - n\bar{h}^2)}} \\
C_4 &= \sum_{i,j} \frac{(g_{i,j} - h_{i,j})^2}{g_{i,j} + h_{i,j}} \\
C_5 &= \sum_{i,j} \left\{ g_{i,j} \log \frac{2g_{i,j}}{g_{i,j} + h_{i,j}} + h_{i,j} \log \frac{2h_{i,j}}{g_{i,j} + h_{i,j}} \right\}
\end{aligned}$$

where n is the number of points in the images being compared, and $\bar{g}, \bar{h}$ are the mean intensities of the two images. Here C_3 is the zero mean cross-correlation, C_4 is the χ^2 test for measuring the similarity of two distributions and C_5 is known as the Jeffrey divergence, which measures the similarity of two distributions based on their relative entropy.

Two other similarity measures were also cited by Smith *et al.* One of these measures is the Kolmogorov-Smirnov statistic, which is used in non-parametric statistics to test whether two distributions are identical. The test statistic is basically the maximum absolute deviation between the cumulative distribution functions of the two samples (each image being first converted to a one dimensional density function by scanning along columns). The final similarity measure provided was the "Earth mover distance" [23] which is the minimum total amount of intensity movement required to shift one distribution of grey-levels into the other. For a one dimensional density function, it can be shown that this metric is the equivalent of the integral of the absolute difference between the cumulative distribution functions. In two dimensions, the movement distances are measured more directly and a linear programming problem is solved.

A number of tests of the above similarity metrics were provided by Smith *et al.* Details of the tests were not described very thoroughly. Despite this, the results still strongly indicate two things. Firstly, it seems important to allow sub-pixel displacements of the interest points. In the paper, a bilinear interpolation step was used to estimate the pixel grey levels from subpixel displacements, and this step appeared to reduce the false alarm rate significantly. Secondly, the KLT similarity (along with the "Earth Mover", the Jeffrey divergence and the χ^2 measures) seem to provide much better tracking performance.

3.3.1 Covariance matching and related corner detectors

As has been a theme throughout this section on tracking, an important property of a corner point is that it is more easily tracked that other points. This subsection considers methods for measuring the ability of a given point to be tracked using the covariance as a measure of similarity of two image chips.

A relatively quick method for computing the ability to track points can be made using a number of simplifying assumptions. Firstly, each $N \times N$ square of imagery can be modelled as being produced by an N^2 dimensional stationary multivariate normal distribution. A maximum likelihood estimate for the parameters of this distribution can be calculated using a training set extracted from the first image frame. The covariance of a given image chip with points from this background distribution will then be a monotonic function of the Mahalanobis distance of that chip from the mean of the distribution. Figure 14 shows a map of this Mahalanobis distance for each pixel in the image. As can be seen, this seems to make a good edge detector, but there is no obvious preference towards corner points.

A measure which has a greater emphasis on corners than that shown in Figure 14 can be achieved by relaxing the assumption of stationarity. Instead, it can be assumed that the statistics have a slow spatial variation, so the parameters can be estimated locally based on a relatively small region about each pixels. This should result in local edges having a greater relative contribution to the local image distribution. This means that the Mahalanobis distance associated with these edges should be less, and so the corners would be emphasised. Because a large number of parameters are estimated based only on local

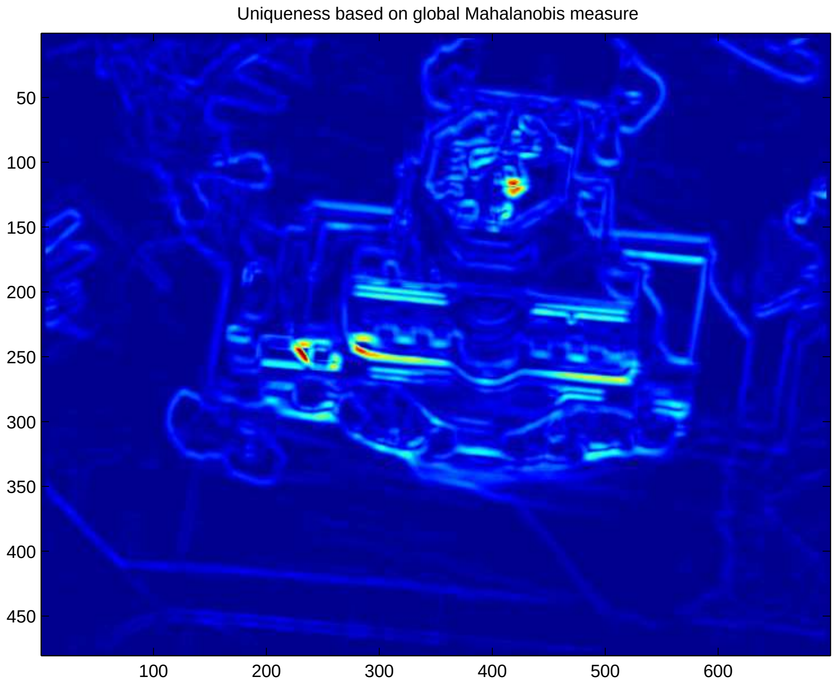


Figure 14: *Uniqueness based on global statistics using a* 9×9 *mask*

data, the rank of the estimated covariance matrix will be too low. This can be ameliorated by the addition of extra training samples which can be simulated by the addition of white noise of some variance σ^2 to the existing samples. This is the same as adding $\sigma^2\mathbf{I}$ to the image covariance matrix. Figure 15 shows maps of the resulting uniqueness measure (over a local neighbourhood of less than 20 pixels) for a number of different choices for the noise variance σ^2.

The above measure of uniqueness has the deficiency of using the non-parametric Mahalanobis distance to measure the uniqueness. It only indicates in an average sense that most of the neighbouring points are dissimilar from the detected points. An individual local pixel might still appear very similar (or could even give a stronger response to a matched filter) without upsetting this average. Perhaps a more accurate measure of uniqueness would be to determine what level of Gaussian white noise σ^2 would be necessary to give a certain percentage (say one in ten thousand) likelihood of a the covariance tracker producing a mismatch in the next image frame. That measure of uniqueness can be calculated using the following steps for each pixel in the image:

- **Local self-match score:** Extract a small image chip $Z_{i,j}$ centered on the target pixel, and compute the self-match score A_{τ_x,τ_y} over a small neighbourhood of translations about zero. In most cases of interest, this will result in a strong peak at the origin.
- **Self-match score variance:** If the next image is a corrupted version of the original image using additive i.i.d noise of variance σ^2, then any value of the match score will have a variance $\sigma_A^2 = \sum_{i,j} Z_{i,j}^2 \sigma^2$, and the distribution should be roughly Gaussian.

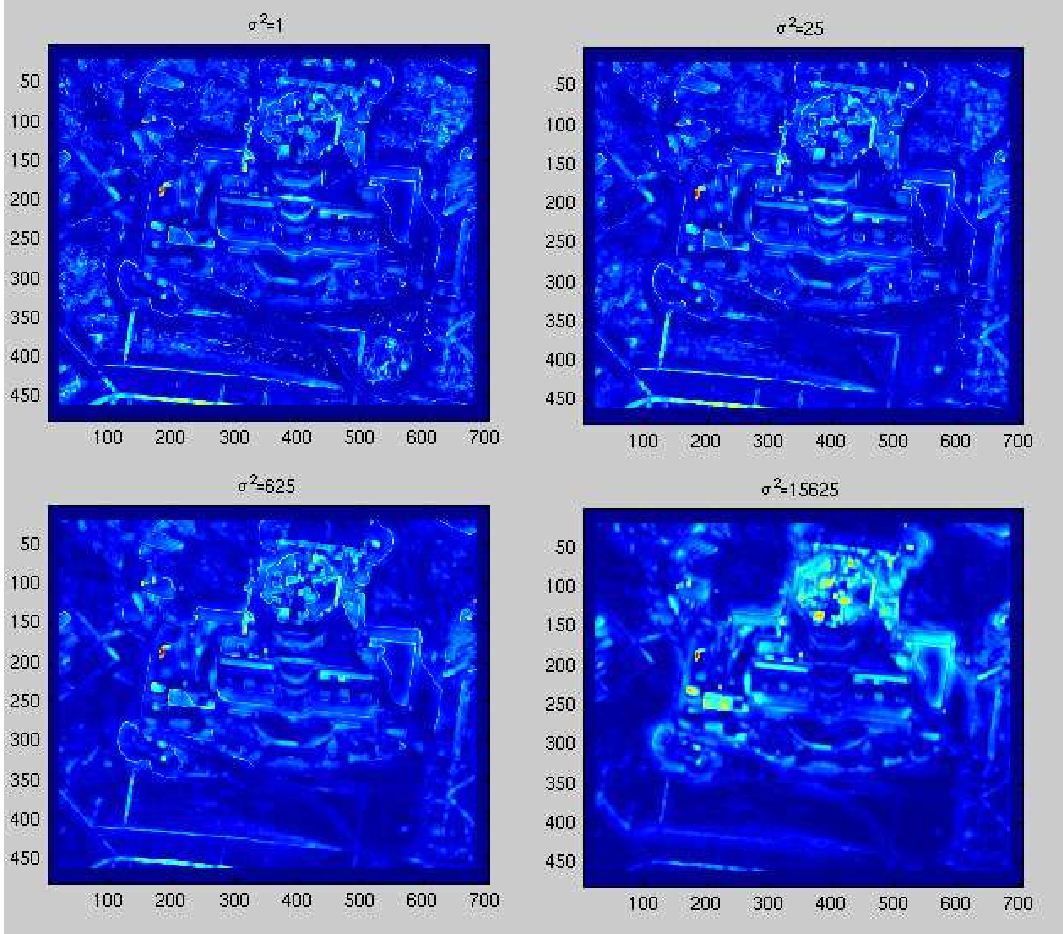


Figure 15: *Maps of the local uniqueness for various levels of image noise σ^2*

Although this will now be spatially correlated noise, to simplify computation it is modelled as uncorrelated.

- **Required noise for false match:** A false match will occur when a point in the neighbourhood of the central pixel has a larger match score. This will occur with probability

$$\alpha = 1 - \prod_{(i,j)\neq(0,0)} \operatorname{erf}\left(\frac{A_{0,0} - A_{i,j}}{\sqrt{2}\sigma_A}\right).$$

  The right hand side will be monotonically increasing with σ^2, so a simple binary search may be implemented to find the amount of noise which gives the required false alarm rate α. In some cases, such a σ cannot be found, in which case σ^2 should be made zero.

The noise required for a 1 in 10000 false alarm rate has been calculated for every pixel in the first Parafield fly-over image. The resulting map is now shown in Figure

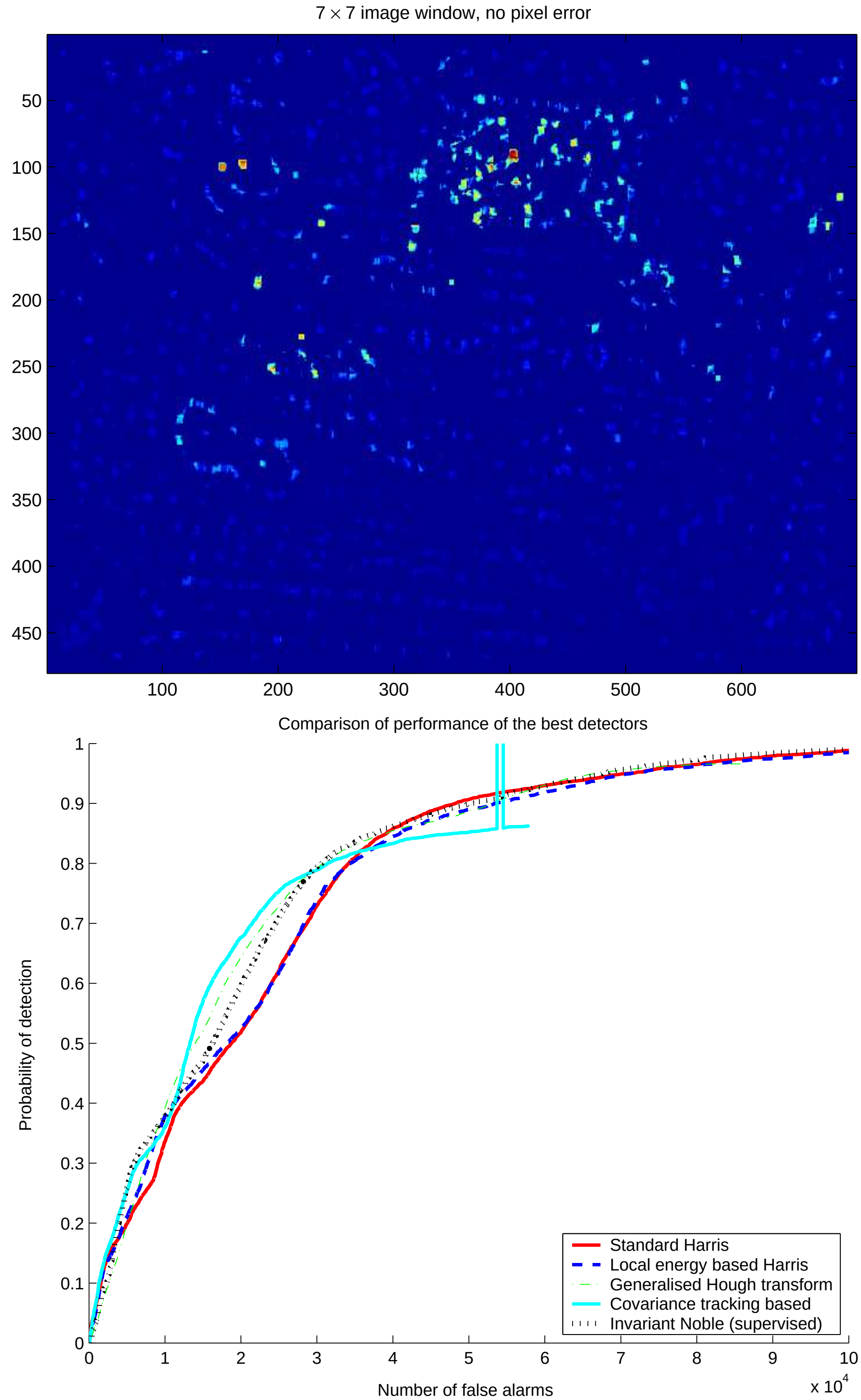


Figure 16: *a) Plot of the noise required to produce a false alarm of* 1 *in* 10, 000 *in the covariance tracker. b) ROC curves comparing the performances of the covariance matching detector and other detectors.*

16. The performance of this detector in finding ground-truthed corners (as described in Subsection 2.7) is also shown in Figure 16, where it has been compared against the best of the detectors previously evaluated. The new detector appears to be the best performing up to a detection probability of about 80 percent, where its performance suddenly collapses.

3.4 Predictive tracking

The tracking algorithms described in the previous subsections all implicitly assume that the image frames are close enough together so that there is virtually no movement of corners between frames. The exact position of the corner is then determined by searching the neighbourhood for similar looking areas of the image. In subsection 3.1, the degree of similarity is the difference in magnitude of the cornerness of the point. In subsection 3.2, the sum of the squared difference of the pixel intensities is used, while in subsection 3.3, the statistical correlation between the two images is used. Each of these methods will have some probability of a mis-match. This can be reduced by choosing the features to track so that this is minimised. Another way to reduce this probability is to decrease the size of the neighbourhood that is searched for a match. This can be done by increasing the accuracy with which the position of the corner is estimated in successive frames. This is the topic of the current subsection.

There are several ways in which the feature motion model can be improved. An obvious way is to treat each feature independently, and estimate its current velocity and acceleration. The position of the corner in later frames can then be predicted using either a constant velocity or constant acceleration model. Both of these models, as well as the zero velocity case, have been investigated by Tissainayagam and Suter [29]. In this paper, it was assumed that a number of points were consistently detected in the frames, and that there were a number of uniformly random detections produced by clutter. Two measures of tracker performance were then described; the probability of a correct association given that it was correctly labelled in all previous frames, and the probability of a correct association given that it was labelled incorrectly in only the last frame. It was found that for a set of simulated data, that the first probability was highest for the constant acceleration tracker, but that the second measure was best for the zero velocity tracker. Some further comparisons were made using a set of real video sequences, each of about 10-50 frames. In all of these experiments, the constant velocity tracker was found to be best. No useful conclusions concerning the trackers were provided in the paper.

The Kalman filter [9] is probably the most frequently used method in tracking applications. The primary assumption behind the model is that at any given time, there is a set of unknown states $\mathbf{x}_k$, and that at the next time interval the states will be updated according to the linear model

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{C_1}\mathbf{n}_{1,k}$$

$\mathbf{n}_k$ is a vector of uncorrelated Gaussian noise, and $\mathbf{C_1}$ is a covariance matrix. Similarly, it is assumed that there is an observed output relating to the hidden states, given by

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{C_2}\mathbf{n}_{2,k}.$$

The matrices $\mathbf{A}$ and $\mathbf{H}$ as well as the covariance matrices and the observations $\mathbf{z}_k$ are assumed to be known constants (although in practice, they could vary). The Kalman filter equations solve for the unknown states and the covariance of the residual $\hat{\mathbf{x}}_k - \mathbf{x}_k$ in a recursive manner consisting of two steps. The first is a predictive step, which estimates the state of the system at the next time step given the estimates at the current time. The second step is a measurement update, which updates the state estimates at the next time based on the observed measurement at that time. When all of the assumptions of the model are satisfied, the Kalman filter method eventually converges to a least squares optimal solution for the state of the system.

For corner tracking applications, a Kalman filter would generally be applied to individual features separately. The state model used would be either a constant velocity or constant acceleration model. The major difference between the Kalman filter predictor and the constant velocity or constant acceleration trackers discussed earlier is the way in which the parameters are estimated. The trackers discussed earlier would use either two or three previous estimates to estimate the current velocity and acceleration of each corner within an image. This means that any measurement error could produce large errors in the predictions of the feature positions in the next frame. The Kalman filter however introduces a smoothing term, to reduce the effects of these fluctuations.

For the application of associating feature points in video sequences, the Kalman filter model is not entirely accurate. One difference is that when the position of a feature is measured, the main source of error will be from confusing it with another part of the image. This labelling error will not, in general, be well represented by a Gaussian distribution. One method for dealing with this source of noise is to use a particle filter [5]. This works in a very similar way to a Kalman filter except that instead of representing the residual error by a covariance matrix, it is represented by a large weighted set of samples from the estimated error distribution. The update equations are also similar, in that they first have a prediction step followed by a measurement based update. The state space updates are performed easily, but the residual distribution update is more difficult. This is because resampling of the distribution is necessary to maintain the total number of particles. The positions of the samples making up the new residual distribution are sampled from a user defined distribution called the importance density. The new weights are then derived from a separate formula. Choosing the best shape of the density used is apparently of crucial importance in determining the performance of the filter.

The Kalman and particle filters can also be extended to track multiple features at the same time. This may be of help in situations where two tracks overlap and it is required to be certain that the resulting tracks do not merge following the intersection. In this case, the dimensionality of the state space would be at least $4N_c$ for tracking N_c corners using a constant velocity tracker (or $6N_c$ for a constant acceleration tracker). This means that for typical structure from motion problems, the size of the state space would be several thousand. This size makes Kalman filtering for this application somewhat unwieldy. The number of particles required to accurately represent a residual distribution increases very rapidly with the number of dimensions, and becomes intractable.

All of the above filters consider each of the features being tracked separately. Usually however, the motion of different features from frame to frame will be highly correlated. A simple way to take this into account would be to assume that a set of points in one frame could be mapped to the next using some unknown affine transformation. This could be estimated easily in a least squares sense from the two preceding frames, and then applied to estimate the feature positions in the next frame. An even more extreme method could assume a projective transform between frames. In this case, structure from motion code could be applied to all of the previous frames, to obtain a set of point scatterers in 3D and a set of camera matrices and feature center of masses. The camera matrix and feature center of mass could then be estimated in the next frame, and applied to the structure to produce a set of predicted positions.

3.5 Track termination

The previous subsections have described methods for determining the most likely track of a feature from one frame to the next. Regardless of the sophistication of the tracker however, sometimes track on a feature will be lost. This may be due either to intrinsic properties of the tracker or obscuration of the feature so that it no longer appears in the image. In either case, it is useful to be able to terminate the track, so that incorrect information is not used in later processing.

All of the tracking methods described previously rely (perhaps implicitly) on some measure of similarity between image fragments in consecutive frames. NNDA effectively uses difference in image corner strength, as measured by some detector, as the indication of the similarity. The KLT uses mean squared difference in intensity, and a number of other frequently used similarity methods are described in Subsection 3.3. Therefore, the easiest method to use in deciding whether a track should terminate would be to use a threshold on the similarity. One example of this type of track termination condition is X34, described by Tommassini *et al.* [31]. Here, if ϵ_i is the measure of the dissimilarity between the feature in the ith frame and in the 1st frame, then the track is terminated when

$$|\epsilon_k - \underset{i}{\mathrm{med}}\, \epsilon_i| < 5.2 \left(\underset{i}{\mathrm{med}} \left\{ |\epsilon_i - \underset{j}{\mathrm{med}}\, \epsilon_j| \right\} \right),$$

where med is the median. The paper by Tommassini *et al.* only used X34 in the context of the KLT detector, so ϵ_i was the sum of squares of the residual error. The same process can, however, be applied to the other measures of similarity discussed previously. This tracking method suffers from the disadvantage that for long image sequences, the shape of the image fragment is likely to change slowly. This will either cause the track to terminate early due to a large difference from the first frame, or will cause the mean absolute deviation estimate to become too large, and the track will be less likely to terminate when it should. To reduce these problems, some sort of adaptive process should be used which would reset the first frame used in the calculation every so often.

A different track termination condition was given by Smith *et al.* [26]. The paper refers to this as the median flow method, and it is based on the assumption that the motion flow field from one frame to the next will be continuous. For a given feature that is being tracked, a set of k neighbouring features are found and the median motion of each of their motion vectors is found. Since these motions will be 2D vectors, the median cannot be computed in a traditional sense. Smith *et al.* define the median angle, which is the mean of the most tightly bunched group of n motion angles. The median length is defined similarly. A track is only continued if a feature's motion is within a certain threshold of both the median angle and the median length. The values of k, n and all of the thresholds need to be chosen by the user. The results given in the paper suggest that this method can reduce the number of false matches by a factor of two. Due to the limited description of the experimental technique, it is not certain that the parameters chosen would work well for all types of data. Also, points on the edges of buildings would have ground points as neighbours within the image. These two sets of points would move differently. As a result, although the percentage of false termination of tracks might be small (as reported in the paper), it might preferentially terminate building edge points. Since these are the points most of interest for determining structure from motion, the median method may not be suitable for the problem. A definite recommendation as to the better track termination condition cannot be made without further tests, and might be a topic for a later report.

4 Fusing corner detectors

Evidently, the different corner formulations presented so far have quite different qualities. All of them preferentially detect corners, but the types of false alarms that are detected can be quite different. For instance, most corner detectors will preferentially detect edges, but the Harris detector specifically deemphasises edges. Similarly, intensity based detectors (Harris, Shi-Tomasi, Kitchen-Rosenfeld, etc) will have a bias towards any area of the image with a large intensity gradient, whereas rank based detectors (such as SUSAN) do not. Therefore, it seems sensible that a number of corner detectors could be combined to reduce the number of false alarms. This section of the report describes some preliminary work in this area.

One simple method for fusing corner detectors is post-processing, where one set of models of corners is used to discard points detected using another. The complete implementation of the SUSAN detector uses this type of processing. The first stage of SUSAN computes a corner measure, based on finding a set of pixels with approximately the same intensity as the central pixel (or nucleus). This set of pixels is called the USAN, and is calculated as described in Subsection 2.3 and tested in Subsection 2.7. The next stage measures the centroid of the USAN. When the centroid is close to the origin, the corner angle is relatively large, and the central pixel is more likely to correspond to an edge than a corner. Therefore a threshold is used here to remove pixels for which the centroid is sufficiently close to the origin. Smith and Brady's original paper [27] did not specify how this threshold is chosen. In this report, it was chosen to maximise the performance for the first frame of the Parafield fly-over. Following this, some more false alarms are removed by keeping only the points for which all of the pixels on a straight line, from the nucleus in the direction of the centroid, belong to the USAN. Figure 17 shows the effect of these post-processing operations on the performance of the SUSAN detector. The post-processing operation has also been applied to the Harris detector, without retuning any of the parameters. The SUSAN detector showed the largest improvement due to post-processing, although the performance is still not as good as for the standard Harris detector. The Harris detector also showed some improvement for lower detection rates, but in this case it

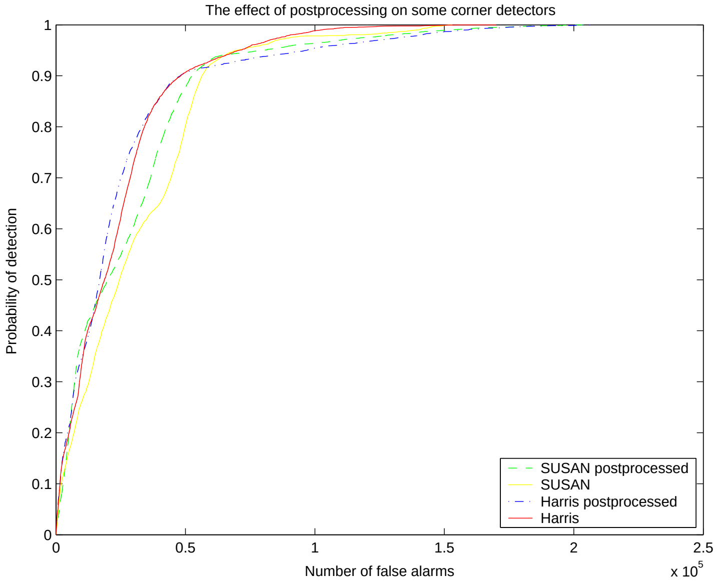


Figure 17: *The effect of SUSAN based post-processing on two corner detectors*

was probably not worth the extra difficulties in implementation and choosing parameters.

Another method for combining corner detectors is based on supervised classification techniques. Here, image chips centered on known corners and false alarms are reduced to feature vectors containing the output from each of the corner detectors. These vectors are then presented to a classifier, which is trained to find a function of the detectors which can be used for discriminating corners from non-corners. An example of such fusion is illustrated in the first diagram of Figure 18, which shows a plot of the output of the Shi-Tomasi and Harris detectors for two classes of points from the first image frame in the Parafield sequence. The blue points are background non-corners, while the red points correspond to manually marked corners, as described in Subsection 2.7. Since these particular detectors give outputs that can vary by many orders of magnitude, they have been scaled using the formula

$$d' = sign(d) \log |d|.$$

By eye, it can be seen that the corners, in general, seem to be confined to almost a straight line. Therefore, the line with this direction $(0.5317, 1)$ as the normal should be able to discriminate corners from non-corners than a line parallel to one of the coordinate axes. This is confirmed by the curve in the second diagram of Figure 18, which shows the performance of the combination of Harris and Shi-Tomasi detectors. As expected, it performs better than either individual detector.

When trying to combine more than two classifiers, it becomes necessary to use automated methods to discover classification rules. In order to be able to cope with the extreme variability and skewness in the distribution of the magnitude of most corner detectors, it was thought that the most suitable classifier should be independent of change of variables

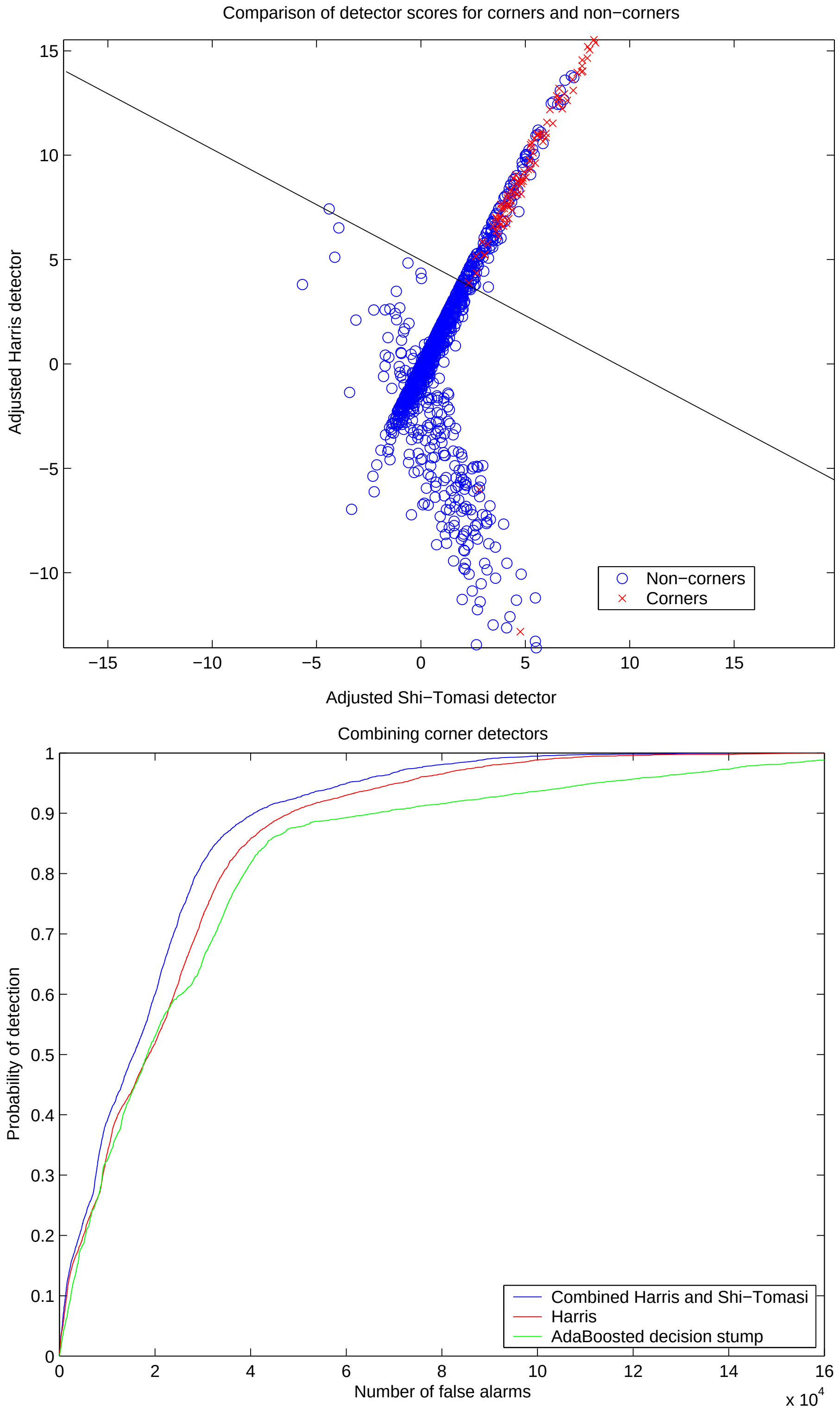
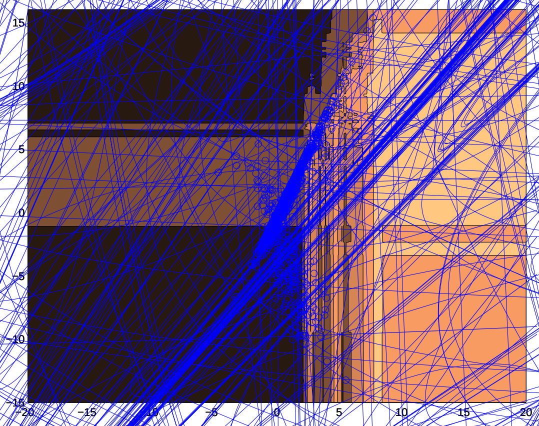


Figure 18: *The effect of fusing corner detectors*

for each detector, as long as the order is preserved. The decision tree is such a classifier, but by itself it is not easy to generate a measure of confidence in the prediction which would be suitable for ROC curve analysis. Therefore, instead of using a single decision tree, a classifier consisting of an ensemble of decision trees, as described in a previous report [2], was produced.

The green curve in the second diagram of Figure 18 shows the performance of an ensemble of decision trees which was used to fuse corner detectors. The curve was obtained using a base classifier, which was a two level decision tree where, for speed, the cuts were chosen to be half way between the means of the corner and non-corner distributions. Each decision tree gave a simple corner/not-corner decision for each point in the training set. The ensemble method was then applied, which reweighted the training points and obtained new base classifiers based on the new weights. The final decision was then a weighted sum of the votes of the individual base classifiers. The reweighting formula and the classifier weights were obtained using the AdaBoost algorithm. More complicated ensemble methods exist, but to get a quick result, only this method was tested. Unfortunately, the resulting fused detector was comparable to the Harris, but shows no obvious advantage to some of the other detectors. It was thought this may have been due to a lack of training data, so the result was repeated with data from the first nine frames being used in the training set. This also did not give an improved result. However, when the ensemble method was applied only to the Harris and Shi-Tomasi detectors, a similar lack of improvement was found, which indicates that perhaps the base classifier is not suitable for solving this particular classification task. Figure 19 shows decision curves obtained by attempt to fuse the Shi-Tomasi and Harris detectors using AdaBoost with a two node decision tree and Fisher's discriminant as base classifiers. Neither automatic classifier fused as well as the manual linear combination. It seems as though these particular classifiers don't work very well when the feature vectors are highly correlated.



5 Conclusions

This report has summarised a number of corner detection and tracking algorithms that have been frequently used for structure from motion work. The ability of each of these corner detectors, as well as several newly developed corner measures, to detect known corner points in real imagery has also been evaluated. Of the tested standard corner detectors, the Harris method consistently gave the best performance. It has also been shown that the performance of the Harris detector can be improved by using a non-Gaussian weighting function. It has not yet been shown whether this improvement will be video specific, or if it is more general.

Of the new detectors, the Generalised Hough transform detector and the covariance tracker based detector appear to have better performance than the Harris detector for a large percentage of corners. The new detectors are, however, slower to calculate, which decreases their potential usefulness. Also, the GHT detector needs an edge intensity threshold to be determined in order to produce good results. It has not yet been determined how this threshold can be calculated automatically. Despite these drawbacks, these detectors are potential substitutes for the Harris detector in structure from motion applications.

The effect of fusing corner detectors has also been examined. By examining a scatter plot, manual fusing of the Shi-Tomasi and Harris detectors was achieved, and the resulting detector was significantly better than each of the detectors considered singly. A number of ways to fuse these two detectors automatically using AdaBoost did not work nearly as well. Automatic fusion of a larger set of 10 corner detectors also resulted in a detector worse than the Harris detector, even though this was one of the detectors being fused. This poor performance may be due to the high correlation between the corner detectors, although more work is needed to establish this for certain, and to produce a work-around for the problem.

The section on tracking has described a number of ways that a particular pixel can be matched between image frames. No simulations have been made to evaluate the performance of the point trackers, but examination of the literature indicates that the commonly used KLT tracker is one of the better options. Several subsections have also been devoted to considering the relationship between tracking algorithms and corner detectors. A measure of the ease with which a point can be tracked is very similar to a corner detector, and several specific examples were analysed. The most successful of these was based on the covariance tracker.

Although this report has provided a great deal of information concerning corner detectors and tracking, there are still numerous avenues of research still unexplored. For instance, corner models have only been mentioned in passing. The typical model considered in this report is a sharp corner in a linear edge between two regions of uniform intensity. Subsection 2.7, described a detector which was trained on such a simulated data set, and although it improved detection for a significant fraction of corners, the detection of the remaining true corners became worse. This must be due to a discrepancy between the actual corners and the corner models. This has not yet been investigated.

There are also areas of further research in the relation between corner tracking and detection. It has been described how the ease of tracking a given pixel is related to how much like a corner it is. The measure of ease of tracking used in this report was to find the level of white noise required to give a particular false alarm rate. Some simulations were given for an NNDA tracking example, as well as for the KLT tracker, but these were based on crude approximations. Finding more accurate measures for these might result in better corner detectors. Also, a survey on optical flow techniques found that phase congruency techniques and a match measure based on the earth-mover measure gave better registration than the KLT tracker. A more detailed look at these tracking methods, as well as a measure of the ability of these methods to track a given pixel, could also result in improved corner detection. It is also of interest to examine different ways of measuring the performance of a tracker, such as setting an image noise level and using the probability of a false match.

Another area of research, which is only mentioned briefly in the current report, is corner localisation. This is implicitly assumed to be done by finding the maximum in a corner map. It is well known however that a good detector (such as the Harris detector) does not always localise the corner very accurately. For this reason, it might be useful to have a separate corner localisation step following detection.

Finally, the work on tracking in this report mostly dealt with its relation to corner detection. The performance of each tracker was compared only implicitly through the detection rates of the associated corner detectors. Future work might compare the performance of trackers for structure from motion in more detail. This would also allow an examination of associated areas such as track termination criteria.

References

1. J.Barron, D.Fleet and S.Beauchemin, “Performance of optical flow techniques,” International Journal of Computer Vision, Vol.12, No.1, pp.43–77, 1994.

2. T.Cooke, “ Second report on SAR image analysis in maritime contexts : Low Level Classification”, CSSIP Technical Report, 2005.

3. E.R.Davies, “Machine Vision,” Elsevier, London, 2005, Chapter 14.

4. D.Fleet and A.Jepson, “Computation of component image velocity from local phase information,” International Journal of Computer Vision, Vol.5, pp.77-104, 1990.

5. N.Gordon, D.Salmond and A.Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” IEE Proceedings-F: Radar and Signal Processing, Vol.140, pp.107–113, 1993.

6. C.Harris and M.Stephens, “A combined corner and edge detector,” Proceedings of the 4th Alvey Vision Conference, University of Manchester, pp.147–151, 1988.

7. F.Heitger, L.Rosenthaler, R. von der Heydt, E.Peterhans and O.Kuebler, “Simulation of neural contour mechanism: from simple to end-stopped cells,” Vision Research, Vol.32, No.5, pp.963–981, 1992.

8. B.M.Hill, “A simple general approach to inference about the tail of a distribution,” Annals of Statistics, Vol.3, pp 1163–1174, 1975.

9. R.Kalman, “A new approach to linear filtering and prediction problems,” Transactions of the ASME - Journal of Basic Engineering, pp.35–45, March 1960.

10. T.Kaneko and O.Hori, “Feature selection for reliable tracking using template matching,” Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Vol.1, pp.796–802, 2003.

11. L.Kitchen and A.Rosenfeld, “Gray-level corner detection,” Pattern Recognition Letters, Vol.1, No.2, pp.95–102, December 1982.

12. T.Lindeberg, “Feature detection with automatic scale selection,” International Journal of Computer Vision, Vol.30, No.2, pp.79–116, 1998.

13. D.Lowe, “Object recognition from local scale-invariant features,” Proceedings of the International Conference on Computer Vision, Corfu, September 1999.

14. B.Lucas and T.Kanade, “An iterative image registration technique with an application to stereo vision,” Proceedings of the DARPA Imaging Understanding Workshop, pp.121–130, 1981.

15. L.Martinez-Fonte, S.Gautama, W.Philips, “An empirical study on corner detection to extract buildings in very high resolution satellite images,” IEEE-ProRisc 25-26 November 2004, Veldhoven, The Netherlands, Proceedings of ProRisc 2004, pp.288-293.

16. J.H.McClellan, “The design of two dimensional digital filters by transformations,” reprinted in Chapter 18 of “Two-dimensional Signal Processing”, S.K.Mitra and M.P.Ekstrom eds., Dowden Hutchinson and Ross, Stroudsburg Penn., 1978

17. K.Mikolajczyk and C.Schmid, “Scale and affine invariant interest point detectors,” International Journal of Computer Vision, Vol.60, No.1, pp.63–84, 2004.

18. H.Moravec, “Towards automatic visual obstacle avoidance,” In Proceedings of the 5th International Conference on Artificial Intelligence, p.584, 1977.

19. J.A.Noble, “Finding corners,” Image and Vision Computing, Vol.6, No.2, pp.121–128, May 1988.

20. K.Paler, J.Föglein, J.Illingworth and J.Kittler, “Local ordered grey-levels as an aid to corner detection,” Pattern Recognition, Vol.17, No.5, pp.535–543, 1984.

21. B.Robbins and R.Owens, “2D feature detection via local energy,” Image and Vision Computing, Vol.15, pp.353–368, 1997.

22. P.Rockett, “Performance assessment of feature detection algorithms: A methodology and case study on corner detectors,” IEEE Transactions on Image Processing, Vol.12, No.12, pp.1668–1676, December 2003.

23. Y.Rubner, C.Tomasi and L.Guibas, “A metric for distributions with applications to image databases,” Proceedings of the 1998 IEEE Conference on Computer Vision, Bombay, India, 1998.

24. C.Schmid, R.Mohr and C.Bauckhage, “Evaluation of interest point detectors,” International Journal of Computer Vision, Vol.37, No.2, pp.151–172, 2000.

25. J.Shi and C.Tomasi, “Good features to track,” Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR’94), Seattle, June 1994.

26. P.Smith, D.Sinclair, R.Cipolla and K.Wood, “Effective corner matching,” Proceedings of the 9th British Machine Vision Conference, H.Lewis and M.Nixon eds., Southampton, Vol.2, pp.545–556, September 1998.

27. S.Smith and J.Brady, “SUSAN - a new approach to low level image processing,” Journal of Computer Vision, Vol.23, No.1, pp.45–78, May 1997.

28. P.Tissainayagam and D.Suter, “Assessing the performance of corner detectors for point feature tracking applications”, Image and Vision Computing, Vol.22, pp.663–679, 2004.

29. P.Tissainayagam and D.Suter, “Performance prediction analysis of a point feature tracker based on different performance models,” Computer Vision and Image Understanding, Vol.84, No.1, pp.104–125, October 2001.

30. C.Tomasi and T.Kanade, “Shape and motion from image streams: A factorization method – Part 3: Detection and tracking of point features,” Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

31. T.Tommasini, A.Fusiello, E.Trucco and V.Roberto, “Making good features track better,” IEEE Conference on Computer Vision and Pattern Recognition, pp.178–183, 1998.

A Genetic algorithm implementation

Genetic Algorithms (GAs) were inspired by Darwinian evolution, which postulated that the characteristics of animals evolved by beneficial mutations, which increased the fertility of the animal possessing them. As a result, the new characteristic would spread quickly to the rest of the population. Similarly, in a GA a population of individuals is described by a set of genes which undergo mutation and cross-over operations. The particular genes that are chosen to form the next generation are chosen randomly, but preference is given to those individuals which produce a larger value of some reward function, which is linked to the problem to be solved. It is claimed that no two people have implemented the same genetic algorithm code. For this reason, the exact method of implementation has been described in this appendix to allow results to be repeated.

One of the first requirements for a genetic algorithm is a genotype (*i.e.* the representation of the system parameters to be used by the genetic algorithm for cross-over and mutation). It is quite common to use binary strings to represent each of the real number parameters. For these results however, each parameter was represented by a real number.

Once the genotype has been chosen, methods for cross-over and mutation of the genotypes must be determined. When the genotype consists of binary strings, one or two point cross-over is usually performed. One point cross-over is where a position in the sequence is randomly selected, and a new genotype is created using the genotype of the first parent from before that point, and the genotype of the second afterwards. More frequently used is two-point cross-over where the first parent provides the genotype between two randomly chosen positions, and the other parent provides the rest. For this implementation, cross-over is a simple arithmetic average of two quantities, while the mutation was the addition of a mean zero random variable. The mutation rate needs to be specially chosen so that, on the one hand, when the individuals are near the optimum, they are not driven away by mutation, and on the other, that when the fitness function is flat, the diversity of the population should increase. For this reason, the mutation variance for a new individual has been chosen to be proportional to the difference in the genotype v of the two parents, so

$$\sigma^2 = \beta(v_1 - v_2)^2/4$$

where β describes the rate of increase in diversity for a flat (on average) reward function. This value can strongly affect the performance of the genetic algorithm, since a larger value can help speed the convergence, with the disadvantage of a higher uncertainty in the final solution. In these simulations, β has been chosen to be 1.2.

The remaining requirements for a genetic algorithm are the reward function, and the method for choosing parents for the next generation. The reward function will of course be problem dependent, but it would be useful to have the same convergence behaviour for the algorithm for any monotonic function of the reward function. As a result, the choice of parents from the population will be made with a probability which is dependent only on the order after they have been ranked.

The literature on genetic algorithms describes a number of ways to define the parenthood probability as a function or rank. The linear function

$$p_i = \frac{1 - \alpha i}{N - \alpha N(N+1)/2} \tag{A-1}$$

is frequently chosen, where α is a constant between 0 and 1. Rather than just using this function blindly, it was investigated whether a better function was available. This is also a stochastic optimisation problem, which can be solved using a GA where the reward function is based on the output of another genetic algorithm. In this case, the GA in the outer loop would be updating a population of probabilities (which for a colony size of 100 would be a vector of 100 positive real numbers summing to one). Since the optimal rank probabilities are not known in advance for this outer loop, the above linear function with $\alpha = 0.01$ was used. The inner GA then uses the probabilities described by the genotype in the outer GA to solve a simple $1D$ optimisation problem. This $1D$ problem used a quadratic function with additive white noise as the reward function, and the inner GA was run for 10 iterations. The mean error between the known optimum position and the position of each individual in the population was then used as a reward function for the outer GA.

When the outer GA was iterated 100 times, the resulting solution for the probabilities versus rank was quite noisy. This is obviously due to the fact that each of the individual quantities can vary significantly, without strongly affecting the convergence of the GA. A

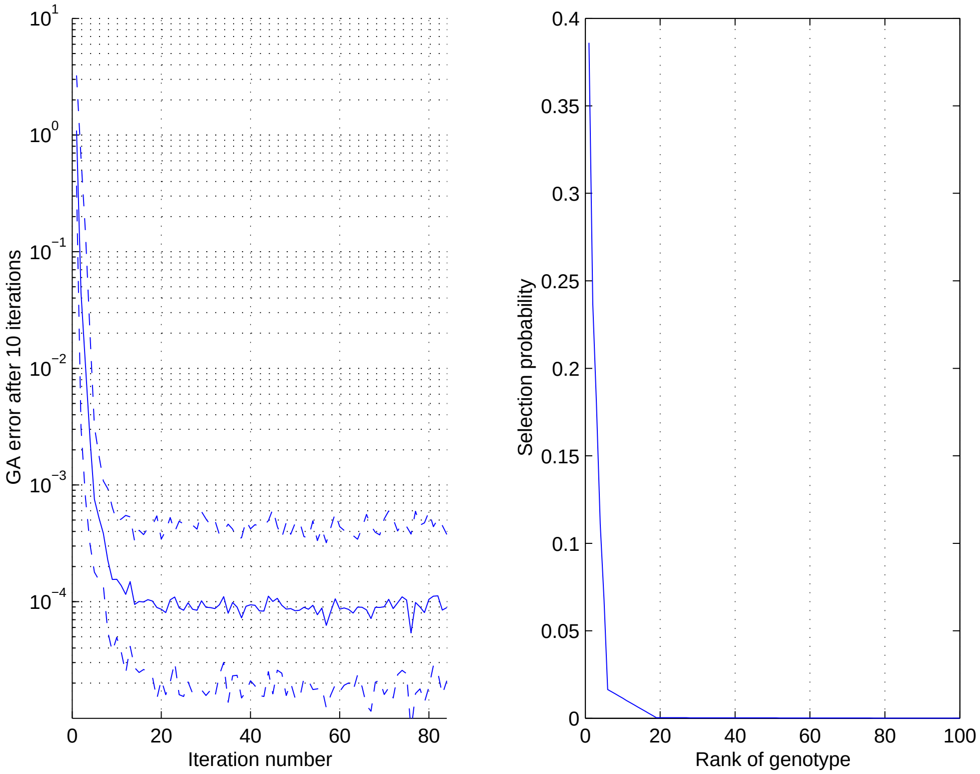


Figure A-1: *Calculation using monotonic piecewise linear constraints of best rank vs selection probabilities for a GA solving a 1D problem.*

more accurate function could be achieved after many more iterations, or by using extra constraints such as continuity or monotonicity. Therefore, to reduce the variance, the selection function was modelled again using a continuous monotonic piecewise function consisting of seven linear segments. This reduced the number of parameters from 100 to 9 (the positions of all of the tie points but one were specified). The resulting selection probabilities are now shown in Figure A-1.

This selection function almost completely ignores the lower rated members of the population. This is presumably because the problem is only one dimensional, so a very small population contains sufficient diversity to be able to solve the problem. To test this, the dimensionality of the problem solved by the inner GA was increased to 20, and the outer GA was restarted. The resulting set of new weights is shown in Figure A-2 where, as expected, the selection weighting does not drop quite so rapidly, so lower rated individuals are more highly valued. It seems that equation (A-1) might still be used, with a value of α which is proportional to the dimensionality of the problem. To test this hypothesis more carefully however, a larger number of different types problem would need to be considered.

The final consideration for a genetic algorithm is how to replace existing individuals. While more complicated systems (such as tournaments) may be used to decide which individuals to dispose of, the implementation here is based on generations, where all of the members in the previous generation are replaced by new individuals. An exception has been made in the case of the fittest individual which is retained unchanged between

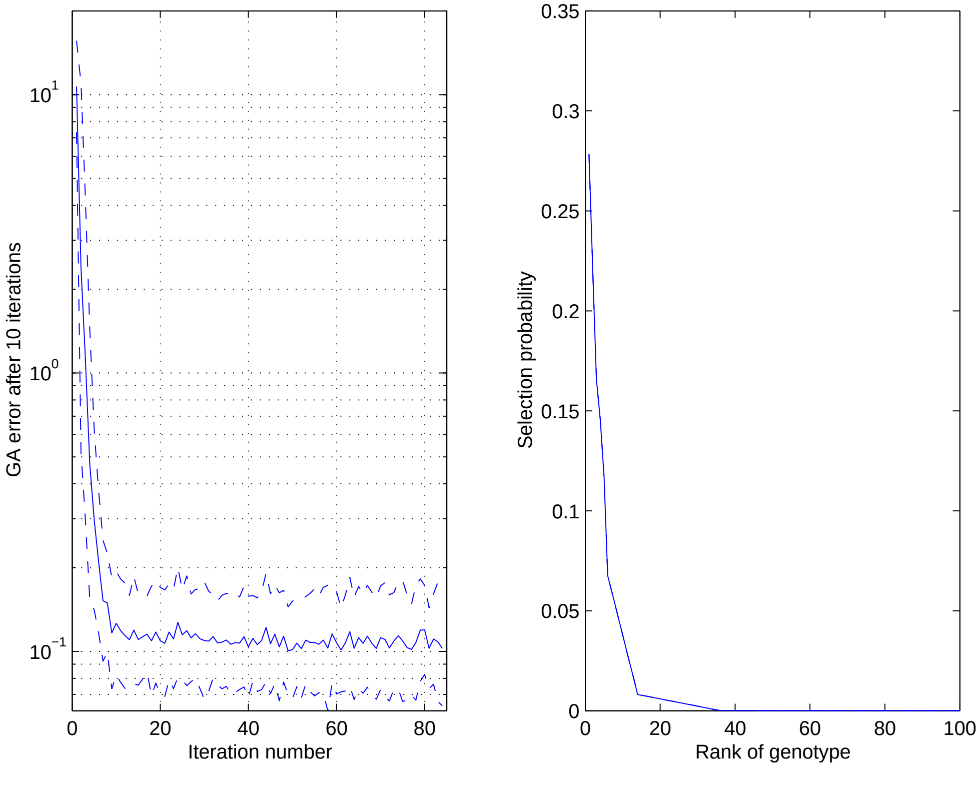


Figure A-2: *Calculation using monotonic piecewise linear constraints of best rank vs selection probabilities for a GA solving a* 20 *dimensional problem.*

generations. This is because sometimes the global maximum can lie on a peak with a very small domain, and if one of the individuals from one generation lands here, there might still be very little chance that any of its offspring would belong to the same peak.

DISTRIBUTION LIST

Detection and Tracking of Corner Points for Structure from Motion

Tristrom Cooke and Robert Whatmough

	Number of Copies
DEFENCE ORGANISATION	
Task Sponsor	
DGICD	1 (printed)
S&T Program	
Chief Defence Scientist	1
Deputy Chief Defence Scientist Policy	1
AS Science Corporate Management	1
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Data Sheet
Counsellor, Defence Science, Washington	Doc Data Sheet
Scientific Adviser to MRDC, Thailand	Doc Data Sheet
Scientific Adviser Joint	1
Navy Scientific Adviser	Doc Data Sheet and Dist List
Scientific Adviser, Army	Doc Data Sheet and Dist List
Air Force Scientific Adviser	Doc Data Sheet and Exec Summ
Scientific Adviser to the DMO	Doc Data Sheet and Dist List
Information Sciences Laboratory	
Chief, Intelligence, Surveillance and Reconnaisance Division	Doc Data Sheet and Dist List
Research Leader, Imagery Systems	Doc Data Sheet and Dist List
Head, Image Analysis and Exploitation	2 (printed)
Ronald Jones	1 (printed)
Tristrom Cooke	4 (printed)
Robert Whatmough	6 (printed)
David Booth	1 (printed)
Robert Prandolini	1 (printed)
System Sciences Laboratory	
Leszek Swierkowski	1 (printed)
DSTO Library and Archives	
Library, Edinburgh	2 and Doc Data Sheet

Defence Archives	1 (printed)

Capability Development Group

Director General Maritime Development	Doc Data Sheet
Director General Capability and Plans	Doc Data Sheet
Assistant Secretary Investment Analysis	Doc Data Sheet
Director Capability Plans and Programming	Doc Data Sheet
Director General Australian Defence Simulation Office	Doc Data Sheet

Chief Information Officer Group

Head Information Capability Management Division	Doc Data Sheet
AS Information Strategy and Futures	Doc Data Sheet
Director General Information Services	Doc Data Sheet

Strategy Group

Director General Military Strategy	Doc Data Sheet
Assistant Secretary Governance and Counter-Proliferation	Doc Data Sheet

Navy

Director General Navy Capability, Performance and Plans, Navy Headquarters	Doc Data Sheet
Director General Navy Strategic Policy and Futures, Navy Headquarters	Doc Data Sheet
Deputy Director (Operations) Maritime Operational Analysis Centre, Building 89/90, Garden Island, Sydney	Doc Data Sheet and Dist List
Deputy Director (Analysis) Maritime Operational Analysis Centre, Building 89/90, Garden Island, Sydney	

Army

ABCA National Standardisation Officer, Land Warfare Development Sector, Puckapunyal	Doc Data Sheet (pdf format)
SO (Science), Deployable Joint Force Headquarters (DJFHQ)(L), Enoggera QLD	Doc Data Sheet
SO (Science), Land Headquarters (LHQ), Victoria Barracks, NSW	Doc Data Sheet and Exec Summ

Air Force

SO (Science), Headquarters Air Combat Group, RAAF Base, Williamtown	Doc Data Sheet and Exec Summ

Joint Operations Command

Director General Joint Operations	Doc Data Sheet
Chief of Staff Headquarters Joint Operation Command	Doc Data Sheet
Commandant, ADF Warfare Centre	Doc Data Sheet
Director General Strategic Logistics	Doc Data Sheet

COS Australian Defence College	Doc Data Sheet

Intelligence and Security Group

Assistant Secretary, Concepts, Capabilities and Resources	1
DGSTA, DIO	1 (printed)
Manager, Information Centre, DIO	1
Director Advanced Capabilities, DIGO	Doc Data Sheet

Defence Materiel Organisation

Deputy CEO, DMO	Doc Data Sheet
Head Aerospace Systems Division	Doc Data Sheet
Head Maritime Systems Division	Doc Data Sheet
Program Manager Air Warfare Destroyer	Doc Data Sheet
CDR Joint Logistics Command	Doc Data Sheet

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy Library	1
Head of Aerospace and Mechanical Engineering, ADFA	1
Deakin University Library, Serials Section (M List), Geelong, Vic	1
Hargrave Library, Monash University	Doc Data Sheet

OTHER ORGANISATIONS

National Library of Australia	1
NASA (Canberra)	1

INTERNATIONAL DEFENCE INFORMATION CENTRES

US - Defense Technical Information Center	1
UK - DSTL Knowledge Services	1
Canada - Defence Research Directorate R&D Knowledge and Information Management (DRDKIM)	1
NZ - Defence Information Centre	1

ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts, US	1
Documents Librarian, The Center for Research Libraries, US	1

INFORMATION EXCHANGE AGREEMENT PARTNERS

SPARES

DSTO Edinburgh Library	5	(printed)

Total number of copies: printed 24, pdf 22

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE Detection and Tracking of Corner Points for Structure from Motion	3. SECURITY CLASSIFICATION Document (U) Title (U) Abstract (U)
4. AUTHORS Tristrom Cooke and Robert Whatmough	5. CORPORATE AUTHOR Defence Science and Technology Organisation PO Box 1500 Edinburgh, South Australia 5111, Australia

6a. DSTO NUMBER DSTO–TR–1759	6b. AR NUMBER 013-476	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE 1st August 2005

8. FILE NUMBER 2005/1067212/1	9. TASK NUMBER INT 04/028	10. SPONSOR DGICD	11. No OF PAGES 49	12. No OF REFS 31

13. URL OF ELECTRONIC VERSION http://www.dsto.defence.gov.au/corporate/reports/DSTO–TR–1759.pdf	14. RELEASE AUTHORITY Chief, Intelligence, Surveillance and Reconnaissance Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved For Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

Corner detection	Feature detection
Image sequences	Three dimensional displays

19. ABSTRACT

This report describes the first stage in solving the structure from motion problem, which is to detect feature points and track them from frame to frame. A number of detectors from the literature, as well as some specially developed detectors, are assessed using a fly-over sequence of Parafield control tower. It is found by this measure that the Harris detector is the best of the conventional detectors, and that two new detectors (the Generalised Hough transform and covariance tracking based methods) appear to give even better results for many cases. Finally, a method for detecting corners by fusing the outputs of numerous detectors is described.